AFOSR 66-0628

AD630792

OPTIMIZATION and STANDARDIZATION of INFORMATION RETRIEVAL LANGUAGE and SYSTEMS

Final Report

Contract AF 49(638)-1194

UNIVAC
DIVISION OF SPERRY RAND CORPORATION
P.O. BOX 500 BLUE BELL PENNSYLVANIA

OPTIMIZATION and STANDARDIZATION of INFORMATION RETRIEVAL LANGUAGE and SYSTEMS

Final Report

Contract AF 49(638)-1194

UNIVAC

DIVISION OF SPERRY RAND CORPORATION
P.O. BOX 500 BLUE BELL, PENNSYLVANIA

28 January 1966

OPTIMIZATION AND STANDARDIZATION

OF

INFORMATION RETRIEVAL LANGUAGE AND SYSTEMS

Earl G. Fossum
Gilbert Kaskey

Final Report under

Contract AF49(638)1194

Prepared for

Directorate of Information Sciences
Air Force Office of Scientific Research
Office of Aerospace Research
United States Air Force
Washington, D. C.

Univac Division
Sperry Rand Corporation
P.O. Box 500
Blue Bell, Pennsylvania

TABLE OF CONTENTS

FIGURES, CHARTS AND TABLES

		Page
Table A-1A.	599 Most Common DDC Descriptors With Field and Group Classifications (In Sequence by Frequency of Usage)	73
Table A-1B.	599 Most Common DDC Descriptors With Field and Group Classifications (In Sequence by Frequency of Usage)	74
Table A-1C.	599 Most Common DDC Descriptors With Field and Group Classifications (In Sequence by Frequency of Usage)	75
Table A-1D.	599 Most Common DDC Descriptors With Field and Group Classifications (In Sequence by Frequency of Usage)	76
Table A2.	Pair Associations Among the 599 Most Common DDC Index Terms Classified by Number of Occurrences	77
Table A-3.	Pair Associations Among the 599 Most Common DDC Descriptors, Classified by Number of Different Pairs and Total Occurrences.	78
Table A-4.	Pair Occurrences as a Percentage of Total Individual Descriptor Usage, 599 Most Common DDC Descriptors.	79
Table A-5.	Pair Occurrences of the 50 Most Frequently Used DDC Descriptors (Selected Summary Data). . . .	80
Table A-6.	Summary Statistics of 599 Most Common DDC Descriptors, Classified by Field of Interest . .	81
Table A-7.	Summary of Pair Associations Classified by DDC Groups to Which Descriptors are Assigned . . .	82
Table A-8.	599 Most Common DDC Descriptors: Occurrences of Pair Associations Within One Group or One Field-of-Interest.	83
Table A-9.	Pair Associations of 599 Most Common DDC Descriptors, Classified by Field-of-Interest Assignment of Each Member of Pair.	84
Table A-10.	Number of Descriptor Pair Associations AB, Classified by Frequency of Usage of Descriptor A . .	86

OPTIMIZATION AND STANDARDIZATION

OF INFORMATION RETRIEVAL LANGUAGE AND SYSTEMS

SUMMARY

The studies described in this report have been aimed primarily at analyzing the organization of data files in the document retrieval application, these being contained in Part I. As a byproduct of a number of analyses conducted on a sample of 38,402 DDC (formerly ASTIA) documents, many term association statistics have been developed. These are presented in Part II, together with a discussion of the implications of association data on file design and use.

A. ORGANIZATION OF DOCUMENT RETRIEVAL INDEX FILES

One proposed type of index file organization is the Multi-List System, a variation of the conventional list-organized file in which the chains or lists are based upon groups of two or three index terms rather than just one. The implications and effects of this proposal have been investigated by a series of computer programs simulating the establishment and maintenance of the files, using as data base the 600 most common index terms in the DDC sample. The results indicate that a large amount of processing, against an extensive data base, is necessary to accomplish the grouping and that the desired objective is not met--most documents have almost as many groups as index terms and the postulated reduction in lists traversing a given document cannot be realized. It is concluded that the Multi-List System does not offer an efficient approach to the organization of a document retrieval file.

One proposed variation of the document-sequenced file orders it on the lowest index term code included in each document description, rather than in straight accession number order. The intent is to reduce the portion of the file searched by eliminating documents which cannot have term codes included in a request. Although this approach is somewhat preferable to the conventional document-sequence file, evaluation indicates that reduction is not enough to make it an efficient method.

Finally, the list-organized file technique is analyzed and compared with the inverted and document-sequenced files. System requirements which can be met with an inverted file are described, together with those which require access to a document record. Analysis shows that the list-organized file is an amalgamation of the inverted and document-sequenced files. It is concluded that maintenance and use of the two separate files is more efficient than the list-organized technique when requirements cannot be met by the inverted file alone. A technique for the optimum detail organization of the two files, by which both actual computing and over-all elapsed processing times can be minimized, is described.

B. TERM ASSOCIATIONS IN DOCUMENT RETRIEVAL

The documents in the DDC sample generate a large number of different pair associations of index terms, most of which occur only one or two times. In general, individual terms form many different pairs and the number increases with total frequency of usage. Terms within one DDC thesaurus group have a high probability of forming pairs and these tend to occur frequently. A lesser tendency, still pronounced, is observed for terms within one field of interest. However, 85% of all pairs involve terms in two different fields. There is no pronounced evidence that index term usage can be predicted upon, or is highly correlated to, the structural hierarchy of the thesaurus. A number of tables summarizing pair association data in the sample are included.

The significance of the high percentage of pairs which occur only a few times is questioned, whether or not such occurrences statistically can be interpreted as representing more than random associations. Some implications are discussed of using associations involving terms of broad scope or wide applicability. It is considered that there is potential application of using relationships implicit in the hierarchal structure of a thesaurus, both in processing search requests and in aiding the describing of documents by such techniques as "lowest level indexing."

Analysis of the DDC data indicates that the use of only a few hundred documents as data base for term association studies generates relationships not representative of the library as a whole. Conclusions derived from these small samples can be highly misleading, particularly if the documents are limited to one subject area. It is believed that meaningful studies require a data base of at least several thousand documents.

The use of term associations is considered to have definite potential in document retrieval. However, the determination of significant associations, the use of thesaurus-implicit relationships in both indexing and searching, and the processing techniques and requirements for incorporating term associations into an operative system, all are deemed to be areas for further investigation.

I. ANALYSES INTO METHODS OF INDEX TERM FILE ORGANIZATION

In an IS&R application, documents are described by a variable number of index terms. Usually, the describing terms are taken from a controlled thesaurus of allowable terms, with their definitions, although sometimes an uncontrolled thesaurus--equivalent to free-language indexing--is used. In either case, the document numbers and associated index terms must be set up in a file which is the data bank against which search requests are processed.

There are four basic ways in which this document number index file can be organized:

a. Document Number Sequence, in which the document number is the record identifer and the associated index terms comprise the body of the record. Every record in the file must be processed against the logical relationships of index terms in a search request. Although the file is usually set up in document number sequence, other orders are permissible and search requests can be processed against a completely random file.

b. Inverted Sequence, in which the index term is the record identifier and the document numbers in which it appears comprise the body of the record. Processing of a search request requires accessing only the index terms it contains, the document numbers pertinent being selected on the basis of the logical relationships connecting the terms of the request. Inverted-sequence files usually are set up in sequence on index term identifying numbers.

c. Document Number Sequence, List Organized, in which each index term associated with a document is "chained" to another document described by that term. The "chain address" can be either a document or its location in the file. A separate entry table contains the document number or its address (file location) of the first document using each index term. The chain addresses permit traversing all documents containing an index term, each single document specifying another in the "chain." Such a file is said to be "list-organized," each index term comprising a "list" which is entered via the entry table and traced through, document by document, using the chain addresses. A document belongs to as many "lists" as it has index terms. In processing a search request whose index terms are connected by logical "and" relationship, one term is selected and only the documents in its "list" examined to determine if the terms describing each one meet the criteria of the request. If the file is maintained on a random access (mass storage) device, it need not be in document number sequence: the "chain addresses" can jump

back and forth through the total file. If stored on magnetic tape or other sequential access devices, the file is set up in document number sequence and the "chain addresses" jump forward, not backward.

d. Superimposed Coding, in which index terms are denoted by randomly selected codes in a fixed-length field, usually binary, and the document description is created by logical superimposition of the codes for the index terms it contains. Each code may be, for example, five random 1-bits in an 80-bit field; the final superimposed code contains a 1-bit in every position in which any one or more of the constituent index term codes has a 1-bit. This type of code may replace, or be generated in addition to, the detail index terms involved; the record key is the document number. Different combinations of index terms can generate the same superimposed code and, as a result, retrievals may include some nonpertinent documents. The percentage of this "noise" can be kept below any desired level by appropriate selection of field size and number of 1-bits in each code. In this type of file organization, every record must be examined in processing a search request. In most cases, however, a document can be accepted or rejected with many fewer comparisons than are required for the conventional document-sequenced file.

A. COMMENTS ON METHODS OF FILE ORGANIZATION

The second and third types of file organization are those which have been studied most intensively in applying electronic data-processing equipment to IS&R applications. In actual operative systems, the inverted file probably is the most common form of file organization, although some magnetic tape applications use a document-number sequence file. The list-organized file appears to be considered suitable primarily when a mass-storage, random-access device is postulated. It is, however, completely feasible when magnetic tape is used. Superimposed coding has had the least consideration and, in operative systems, appears to be restricted to manual operations with files maintained on edge-notched or punch cards, or similar storage media.

There seems to be general agreement that, of the first three types, the document-sequenced file is markedly inferior to either of the other two. The necessity for inspecting every document record in processing a search request entails a comparison work load (matching index terms against those of the search request) two or more orders of magnitude greater than with either an inverted or list-organized file. This factor normally makes it unattractive for batch processing of search requests using any type of current equipment with multiprocessing capabilities. The other two use much less internal processing time, even though the list-organized file essentially doubles the amount of data to be transferred into the computer memory. In practice, a search through a document-sequenced file almost always is a badly tape-limited computer operation; the index terms in each of the several (one or more) requests must be matched against those of each file record until rejection or acceptance occurs. Even though rejection (the common disposition) frequently occurs fairly early in this matching process, the total comparison time normally is several times longer than the actual tape-to-memory transfer time of a record. For this reason,

there is no particular advantage in storing a document-sequenced file on a mass-storage device rather than on magnetic tape; internal computing, not data transfers, governs the total processing time. In a real-time IS&R application, this type of file organization obviously is inapplicable.

The inverted sequence file has the advantage of a small number of records--one for each index term in the thesaurus. Typically, this is on the order of a few thousand, whereas documents are numbered in the tens of thousands. The records are highly variable in length; some index terms appear in only a single document description while others are used in thousands of them. This type of file has two basic implications in the processing of search requests.

First, the only records examined are those for the index terms in the search request (or requests in batch processing) and the output is limited to a list of document numbers satisfying the request logic. Other index terms used to describe the selected documents are not included and cannot easily be obtained from an inverted file. Their omission removes one possible means for quickly determining document pertinency. Second, all document numbers for an index term must be matched against those carried forward to the current stage of processing. Thus an entire record of several hundred document numbers may have to be scanned to find the "matches" against a relative handful which so far have satisfied the request criteria; this may be repeated for several more index terms.

A list-organized file combines the selective-search advantages of the inverted file with the advantage inherent in the document-sequenced organization of obtaining all index terms associated with selected documents. Its disadvantage is that, for practical purposes, file size is doubled when compared with the other two. Like the document-sequenced file, a document record, once accessed, is accepted or rejected on the spot; there is no carry-over to a subsequent stage of processing. The number of records inspected can be minimized if the entry table to the list for each index term includes its number of occurrences in the file. Assuming logical "and" relationships between terms in a search request, it is easy to determine the one with the fewest occurrences and to examine only the documents in that list. More complex term relationships in a request may require entry to and processing of more than one such list, but each can be the shortest one applicable to a subset of the request terms.

The number of records to be accessed in searching a list-organized file can be no less than the number of occurrences of the least frequently used index term in the request. This is highly variable. Some requests may contain a term used in only two or three documents; in others, the least frequent term may have 50, 100 or even more occurrences, and many records must be examined. Complexity of the search request also can affect the number of record accesses required. Ten terms all connected by logical "ands" can be processed by entering a single list. If a few "or" relationships are present and no common "and" term exists, then two or three lists may be entered. Finally, the minimum number of records accessed is almost directly proportional to the size of the library (file) being searched. The thesaurus of index terms, once established, tends to change rather slowly as documents are added to the file. The frequency of usage of index terms increases, on the average, directly as the number of documents--more usages are recorded

to a relatively constant number of index terms. Thus as the library increases in size, more and more records must be accessed in processing a search request. This characteristic is true even when indexing standards remain unchanged. Major thesaurus revisions or different indexing criteria also affect the contents of the file and the processing of requests against it.

With an inverted file, on the other hand, one record is accessed for each term in a search request. Although the number of terms varies, the maximum typically does not exceed the minimum by more than about 10:1, and a fairly high percentage of requests have close to the average number of terms. In general, the number of records accessed is much smaller than with a list-organized file. However the individual records are longer. Other factors remaining unchanged, the number of records to be accessed does not change as the size of the library increases, but the average record length does grow at a rate proportional to that of the number of documents.

With an inverted file, the amount of data transferred into the computer memory to process a search request is relatively more predictable than with a list-organized file and is not subject to so much variation. Assuming each data element to be one word, it is given by

$$\text{Words Transferred} = T_i(N_i + 1),$$

where

T_i = Number of index terms in the request, and

N_i = Average number of documents in which each term appears.

[The "1" in $(N_i + 1)$ assumes that the index term is one word of the record.] Although individual N's vary widely in value--from one to several thousand--for individual D's, the total, and the average, for typical ranges of search requests are subject to much less variation; the maximum may be on the order of 2-3 times the minimum.

With a list-organized file, the number of words transferred is given by

$$\text{Words Transferred} = N_m(2D_i + 1),$$

where

N_m = Number of documents in which the least frequently used index term appears, and

D_i = Average number of index terms per document.

(In this expression, "$2D_i$" appears because each index term has attached to it the chain address of the next document in the list; this also is assumed to require one word for its representation.) Unless N_m is very small, D_i will closely approximate the average number of terms per document in the entire library and thus is readily predictable. However, as has been observed, N_m is highly variable. An examination of about 200 search requests

has not revealed any conclusive relationship between the number of terms in a search request and the overall frequency of usage of any one of them. In general, it appears that a greater number of terms in a request increases the probability of finding one used fairly infrequently in the library. At the same time, requests with many terms tend to have more complex logical relationships and this increases the probability that several index term lists, and not only one, will have to be scanned in processing a request.

Without comparative analysis, it is not possible to determine which of the two types of file organization requires the lesser amount of data transfers in processing a search request. The list-organized file almost certainly does if N_m is not over 2-3 times as large as T_i, but the exact break-even point is not known.

It appears certain, however, that the number of <u>different</u> records to be accessed is considerably greater with the list-organized file. This factor can become highly important when the file is maintained on a mass-storage, random-access device, particularly if the application is real time. In this case, access to records can be made on a random basis with both list-organized and inverted files. Random access time typically is much longer than data transfer time, even for very long records. Consequently, the total <u>elapsed</u> time to process a search request almost always will be greater with a list-organized than with an inverted file (even though the actual central processor time may be less). The break-even point can be taken, with sufficient accuracy for practical purposes, as the case in which the number of index terms in the request equals the minimum number of documents which must be examined. Usually the latter is considerably greater.

Some proposals have been made to modify the technique of setting up a list-organized file to permit more efficient retrieval. One of these has been examined in detail, with negative results, using as data base the large sample of 38,402 DDC (formerly ASTIA) documents described in [1].

B. <u>ANALYSIS OF THE MULTI-LIST SYSTEM</u>

The Multi-List System [2], [3] is a list-organized file in which each list consists of a set of index terms--three being suggested--rather than having a separate one for each term. Several potential advantages have been cited for this type of file organization: (1) Although the number of lists traversing the file is increased, their average length is reduced and variations in length are much less extreme than in the usual list-organized file; (2) a document belongs to fewer lists and, because fewer chain addresses are needed, file storage requirements are less; (3) file searching is faster, because fewer lists must be examined; and (4) the method of organizing the entry table to the lists may permit eliminating some search requests (no pertinent documents in the library), without examining any list, by utilizing knowledge that two index terms have never been used together in a document description.

1. Mutually Exclusive Attribute Groups and Formation of Lists.

The method of combining three index terms into one list is based upon assigning each term to one of a limited number of attribute groups. In any one attribute group, all its index terms are mutually exclusive; that is, no two terms in the group are used together in a document description. The index terms, then, are said to be assigned to mutually exclusive attribute groups. This array is best illustrated by an example.

Suppose a file consists of records each having nine keys or attributes, each attribute in turn having ten mutually exclusive possible values. An attribute, for example, could be military rank; each record (man) can have only one of the possible values "private," "corporal," and so on up to "general." There are 90 different possible values (or index terms) in the file. These can be denoted in the form "0608" for the 8th value in the 6th attribute column, etc. The mutually exclusive attribute groups then look like this:

1	2	3	4	5	6	7	8	9
0101	0201	0301	0401	0501	0601	0701	0801	0901
0102	0202	0302	0402	0502	0602	0702	0802	0902
0103	0203	0303	0403	0503	0603	0703	0803	0903
0104	0204	0304	0404	0504	0604	0704	0804	0904
0105	0205	0305	0405	0505	0605	0705	0805	0905
0106	0206	0306	0406	0506	0606	0706	0806	0906
0107	0207	0307	0407	0507	0607	0707	0807	0907
0108	0208	0308	0408	0508	0608	0708	0808	0908
0109	0209	0309	0409	0509	0609	0709	0809	0909
0110	0210	0310	0410	0510	0610	0710	0810	0910

If each attribute value is placed in a separate list, there are 90 lists in the file. Some (such as "private" or "ages 20-24") are extremely long, while others (e.g., "general") are short. Also, each record in the file belongs to nine lists and has nine tags.

Now let groups of three columns be combined into a single superfield in which a superkey might consits of one attribute value from each column, as 0104-0202-0307. Each superfield has a possible 1,000 (10 x 10 x 10) of these superkeys, not all of which are present in the file (generals, ages 20-24, earning $40-$49 weekly, probably are nonexistent). If a superkey corresponds to a list, there are at most 3,000 in the file (three superfields of 1,000 superkeys each). Although there now are many more lists traversing the file, the extremely long ones previously existing are broken up into many smaller ones by the grouping of three attribute values into one superkey. The

short lists, of course, are even shorter. Each record now has only three, rather than nine, chain addresses or tags.

Alternatively, a superkey may be created by grouping two or more attribute values from each of the three columns, the superkey now representing a range of values rather than a unique combination. For example, 0401 or 0402 may be combined with either 0501 or 0502 and also with either 0603 or 0604, eight different combinations in one superkey. Each column has five of these pairs of values and a group of three columns has 125 superkeys. The array as a whole has 375 of them, defining 375 lists in the file. With proper ordering of attribute values within a column, the very short single lists can be eliminated by combining them with longer ones and all lists made approximately the same length--possibly a 2 or 3 to 1 maximum variation.

This mutually exclusive attribute group array serves as the entry table to the lists traversing the data file. Each superkey in the array has attached to it the storage location (or other identification) of the first record in the list. A desired superkey in the array can be isolated by standard searching techniques which successively narrow the portion of the entry table in which it lies.

2. Application of Multi-List System to IS&R (Document Retrieval).

In many types of files, some (or all) of the data elements are values of attributes, such as age, salary, years of education, etc. Here the existence of a given entry for an attribute precludes, by definition, any other value for one file record; a person cannot have two different ages at the same time. Thus the entries in the attribute group are mutually exclusive.

The index terms in a document file do not have this type of mutual exclusiveness. Although many--perhaps most--pairs may define concepts which are extremely unlikely to co-occur in a document, it is perfectly possible that, given a library of large enough size, any two terms chosen at random will be used in the same document description. Their mutual exclusiveness is strictly a function of usage and two terms which are exclusive today--i.e., have never been used together in one document--may not be tomorrow. Use of the Multi-List System in an IS&R application requires, then, not only an algorithm to set up the array of mutually exclusive attribute groups initially, but also to reorder it when previously exclusive index terms in one group are used together in one document description. This process of changing terms from one column to another to maintain exclusiveness is called renaming.

It is evident that the minimum possible number of attribute groups is at least as great as the largest number of index terms used in a single document description. The actually realizable minimum may be considerably larger, and most likely is. In the DDC sample, the maximum number of terms in any one document is 21.

In applying the Multi-List System to the total DDC document file, it was believed initially that the 6,000-odd descriptors (DDC index terms) could be arranged into about 30 mutually exclusive attribute groups or columns (subsequently raised to 40), each containing about 200 descriptors. In each column descriptors are grouped into ten sets of about 20 each, one set from each of three columns comprising a superkey covering a range of descriptor code-value combinations. Each group of three columns has 1,000 of these superkeys serving to define lists, or a total of 10,000 lists traversing the documents stored in the Multi-Association Area (the file of document numbers, descriptors and chain addresses).

To examine the feasibility of the Multi-List System, a computer algorithm was developed to set up the array of mutually exclusive attribute groups. A UNIVAC I-II program was written and run against the collection of 38,402 DDC documents available for testing and analysis.

3. Descriptions and Results of Experiments Using DDC Data.

The methodology followed and results of the computer experiment are summarized in the following paragraphs. More detailed descriptions have been reported previously in [3]-[7].

This phase of the study sets up the mutually exclusive attribute group array, using descriptor relationships in the DDC data as basis for the allocations, and is designed to answer two basic questions:

What is the achievable minimum number of groups into which the descriptors can be assigned?

How complex a renaming process is required to retain a minimum number of groups as the introduction of new associations necessitates reordering of the array?

a. Notation. T notation used in these analyses is modified slightly from that in published literature on the Multi-List System. Symbols used are:

D -- The description used to describe a document and consisting of a number of descriptors, denoted by either d_a or $d_{i,j}$.

d_a -- The descriptors in D, $1 \leq a \leq m$. Used when location within the attribute groups is not pertinent or is not known.

c_i -- A column or group in the array of mutually exclusive attribute groups. $1 \leq i \leq f$, where c_f is the last group.

$d_{i,j}$ -- The jth descriptor in the ith column; e.g., $d_{07,12}$ is the 12th descriptor in group 7.

$i.j_n$ -- Alternate form for writing $d_{i,j}$, particularly when it is necessary to differentiate two j's in one column.

$i\text{-}v$ -- Notation used to denote a descriptor in c_i, retaining the descriptor's identity. Typically, v is either the descriptor code itself or its frequency of usage rank. Thus, 11-3860 represents descriptor code 3860 in group 11 and 01-23, the 23rd-ranked descriptor (in the total file) in group 01. Each group normally is sequenced on v, but only the v's in the group are included.

C_i -- The set of descriptors in a column c_i. It is $d_{i,j}$, $j = 1, 2, \ldots, n$.

$D_{i,j}$ -- The set of all descriptors with which a $d_{i,j}$ is associated in use. By definition, none of them can be in c_i.

D_i -- The set of all descriptors associated in use with any one or more of the $d_{i,j}$ in c_i. It is the logical sum of all the $D_{i,j}$ in a c_i.

p -- The number of groups not having a descriptor in D.

c_{K_m} -- An individual group or column not having a descriptor in D. $1 \leq m \leq p \leq f$.

List K -- The c_{K_m} with the descriptors included in each.

$a \cap b$ -- "a is inclusive with (used with) b." a and b can vary in form and may differ. Thus, $i,j_1 \cap i,j_2$ or $i\text{-}v_1 \cap i\text{-}v_2$ means that a single specified descriptor pair is inclusive. $i,j_1 \cap c_b$ means that i,j_1 is inclusive with one or more of the $d_{b,j}$, without specifying which one(s).

$a \cup b$ -- "a is exclusive to (not used with) b." a and b can vary as above.

b. <u>Definition of Renaming</u>. Assume that the mutually exclusive attribute groups have been established; no two descriptors in any one column are used together in a document. Now if a new description D contains two previously exclusive descriptors i,j_1 and i,j_2, it is necessary to move one of them into another column. The Multi-List System proposes these types of renamings:

<u>First-Order Renaming</u>. If there is a column c_k such that $i,j_1 \cup c_k$, then i,j_1 can be moved to c_k, becoming k,j_n. Conditions for exclusiveness are now met by c_i. Similarly, exclusiveness may be maintained by moving i,j_2 to c_k.

Second-Order Renaming. There may be no c_k into which i,j_1 (or i,j_2) can be shifted. If, however, a descriptor k,j_x can be shifted into still another column, thereby making $i,j_1 \cup c_k$, then a double shift will maintain the exclusiveness--i.e., $k,j_x \rightarrow c_t$ ($k,j_x \cup c_t$) and i,j_1 (or i,j_2) $\rightarrow c_k$.

nth-Order Renaming. The above process can be repeated any number of times. Without specifying a value, the Multi-List System recognizes that an upper limit to orders of renaming should be set. If the limit is reached without a successful renaming, it is concluded that the input descriptor is inclusive to every column and consequently can be placed in none of them. At this point, either the number of attribute groups is increased by one or recourse is made to a human monitor.

The basic flow chart for the logical operations required for first and second order renamings is shown in Figure 1. Except for slight changes in notation, this is identical with those presented on pp. 67-68, Part I, Volume I of reference [1]. The chart begins at the point where existence of a conflict within a group has become known. In effect, it includes the basic logical operations required for all orders of renaming; those of third and higher order can be considered as successive applications of the second order renaming process.

During the course of the study it became apparent that another type of renaming was not only possible, but also was necessary to maintain the number of groups at a minimum. This is defined thus:

Second-Degree Renaming. The requisite k,j_x may not exist for successful second-order renaming. However, if there exists also a k,j_y such that (1) $D_k - (D_{k,r} + D_{k,s})$ is exclusive to i,j_1; (2) k,j_x is exclusive to some D_m; and (3) k,j_y is exclusive to some D_p (where p may equal m); then the triple movement $k,j_x \rightarrow c_m$, $k,j_y \rightarrow c_p$ and $i,j_1 \rightarrow c_k$ restores exclusiveness.

nth-Degree Renaming. This follows the above principle, except that n descriptors are moved out of c_k.

In theory, any degree of renaming can occur at any stage of an nth-order renaming.

c. Algorithm Requirements and Practical Limitations on the Computer Experiment. Although Figure 1 is complete for the basic logic of what must be done in descriptor renaming, it is not a computer solution or flow chart of how it is to be accomplished. For purposes of this study, it was first necessary to devise a detailed method which was feasible of operation, in terms of time and cost, on the UNIVAC I-II magnetic-tape processors available for use.

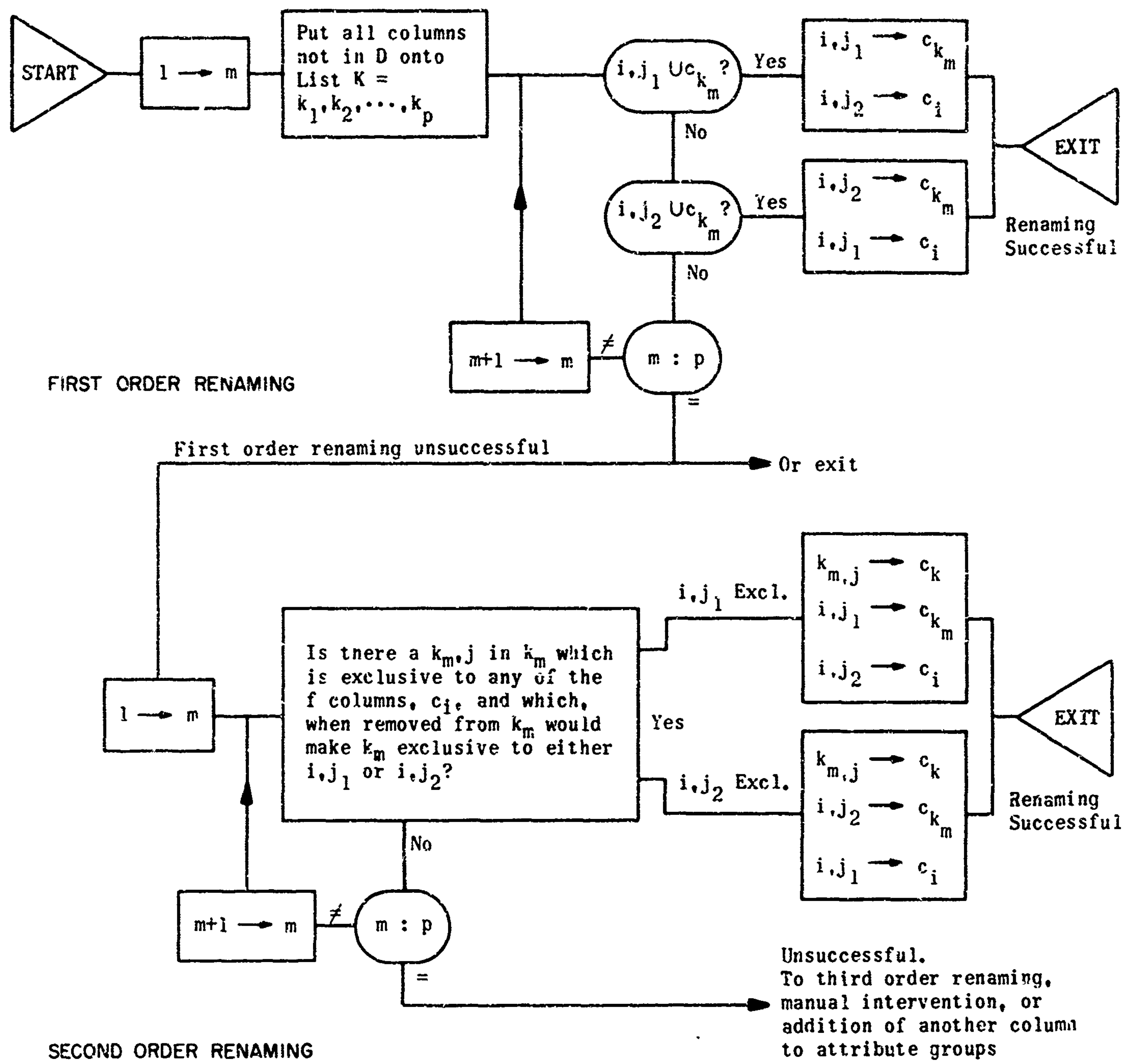


Figure 1

Multi-List System

Logical Flow Charts for Renaming Process
to Maintain Attribute Group Exclusiveness

The study objectives required methods (1) for the inital establishment of the array of attribute groups based upon actual usage of descriptors in a reasonably large "base" document file; and (2) for maintaining the array as new documents are added. Preferably, the same basic machine program should take care of both. The specifications listed below must be met; some of them are framed to reflect the particular characteristics of a tape processing system.

> <u>A file of attribute groups, c_i,</u> and the descriptors C_i within each must be maintained. Within each c_i, the $d_{i,j}$ are ordered in some systematic manner--e.g., in descriptor code or frequency-of-usage sequence.
>
> <u>A cross reference</u> between descriptor name or code and its attribute group must be set up and kept current.
>
> <u>The descriptor set $D_{i,j}$</u> for each $d_{i,j}$ must be established and kept current. So long as documents are not removed from the file, the maintenance procedure need provide only for $D_{i,j} + d_{k,r} \rightarrow D_{i,j}$ when a new association $d_{i,j} \cap d_{k,r}$ is introduced.
>
> <u>The descriptor set D_i</u> must be established and kept current for each c_i. The maintenance procedure must provide for both $D_i + D_{i,j} \rightarrow D_i$ and $D_i - D_{i,j} \rightarrow D_i$ to reflect the effects of the movement (renaming) of a $d_{i,j}$ into or out of c_i.

Computer and cost considerations made it evident that the entire 5540 descriptors in the DDC sample could not be handled. Accordingly, the 599 most frequently used were selected; this includes all descriptors with 72 or more occurrences in the sample file. This choice permitted setting up the descriptor sets D_i and $D_{i,j}$ as 2-block records (UNIVAC I-II blocks of 60 words each) in matrix form, 2-character fields in each record corresponding positionally to the descriptors 001-599 taken in rank-number sequence. The number of co-occurrences of a $d_{i,j}$ with each of the other 598 descriptors can be accumulated readily as 2-digit numbers in the proper positional field (very few pairs occur more than 100 times). The 600th field identifies the descriptor or attribute group to which the record pertains.

Tape-handling considerations also made it clear that computer renaming would have to be restricted to avoid excessive "tape spinning." Thus, renaming was limited to the forward direction of tape movement, equivalent to moving a descriptor only into columns to the right of the one from which it must be moved.

The investigation of aspects not taken care of by the computer program were covered by selecting the 50 most common descriptors and, within them, taking only pairs with five or more co-occurrences. This reduced the amount of data to a volume permitting manual simulation of the algorithm. Some manual simulation also was performed with the 100 most common descriptors.

In all of these studies, descriptors are identified by their rank number based upon frequency of usage in the file; 001 is most frequent and 599, least. A complete list of the 599 descriptors is contained in Table A-1 (Appendix A), in rank number sequence. Each descriptor shows the number of different pairs it forms with the other 598 and also the ASTIA field and group to which it was assigned in 1960-1961 (the period during which most of the documents in the sample were described. It should be noted that field/group assignments have been changed since that time).

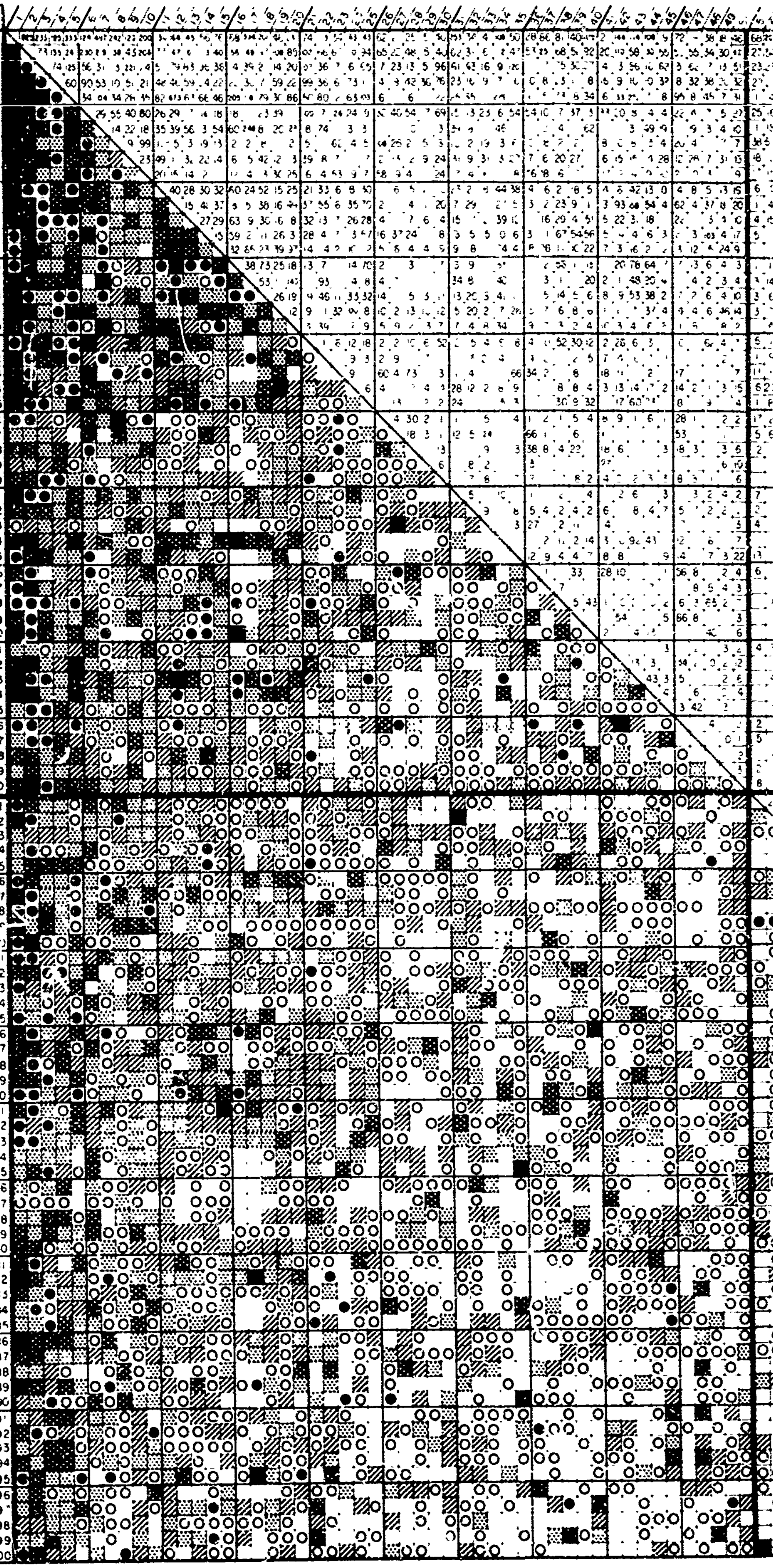
d. Summary Statistics of Pair-Associations Among 599 Most Common DDC Descriptors. These descriptors have 49,306 pair-combinations (twice as many permutations) with 248,425 total occurrences. These constitute 23.6% of the different pairs in the total file and 46.8% of all pair occurrences. On the average, each descriptor is used with 165 of the other 598; the range is from 49 to 579. Table A-2 (Appendix A) summarizes these pair-associations by number of occurrences.

As might be expected, descriptors used very frequently are highly likely to co-occur in document descriptions. Chart 1 depicts the pair-associations among the 100 most common; the lower left triangular matrix is a graphic portrayal of this, with the actual number of co-occurrences of each pair in the upper right portion. Some breakdowns of possible and actually existing pairs are summarized in this table:

Association Type	Possible Combinations	Combinations Occurring	Percentage Occurring
Associations within Ranks 1-50	1,225	1,088	88.8%
Associations of Ranks 1-50 with Ranks 51-100	2,500	1,928	77.1
Associations within Ranks 51-100	1,225	681	55.6
Associations, one member in Ranks 101-599	174,151	45,609	26.2
All Associations, Ranks 1-599	179,101	49,306	27.5%

Although over a fourth of the possible pairs actually exist in the sample, it should be noted that most of them do not occur frequently. In fact, over 40% occur only once and almost 60% only once or twice (see Table A-2).

CAT CODE	DESCRIPTOR	DOC RANK	TOTAL NO OF PAIRS	DIFF PRS	RANK
292	DESIGN	1	8,928	99	1
292	TESTS	2	7,250	99	2
147	MATHEMATICAL ANALYSIS	4	3,555	98	3
292	MEASUREMENT	5	2,584	98	4
114	GUIDED MISSILES	3	4,846	93	5
117	TEMPERATURE	7	2,381	94	6
027	AIRBORNE	6	2,905	90	7
217	PRODUCTION	11	1,689	92	8
292	THEORY	13	1,808	96	9
145	MATERIALS	10	1,921	91	10
292	ANALYSIS	8	1,780	98	11
027	SURFACE-TO-SURFACE	17	2,868	89	12
108	GREAT BRITAIN	14	1,783	98	13
006	STABILITY	16	1,916	94	14
292	EFFECTIVENESS	9	1,625	94	15
183	FLIGHT TESTING	20	2,157	76	16
227	RADAR EQUIPMENT	12	2,461	73	17
208	INSTRUMENTATION	19	1,772	93	18
292	TEST METHODS	15	1,324	97	19
078	COUNTERMEASURES	23	1,694	90	20
216	PRESSURE	27	1,651	89	21
102	DETECTION	24	1,379	88	22
148	MECHANICAL PROPERTIES	28	1,105	66	23
292	TEST EQUIPMENT	25	1,342	93	24
010	CONTROL SYSTEMS	18	1,402	74	25
217	PROCESSING	55	902	82	26
053	SYNTHESIS	31	752	70	27
105	PHYSICAL PROPERTIES	35	896	81	28
209	PHYSIOLOGY	35	377	55	29
117	HEAT TRANSFER	39	931	79	30
076	CIRCUITS	33	1,069	80	31
292	DETERMINATION	37	744	94	32
050	CHEMICAL REACTIONS	40	599	57	33
027	SURFACE-TO-AIR	43	1,479	67	34
247	STRESSES	20	983	69	35
106	POLYMERS	47	818	65	36
020	PROJECTILES	22	868	75	37
006	AERODYNAMICS	26	1,207	76	38
057	COMBUSTION	34	857	71	39
009	JET PLANES	55	1,233	65	40
116	RADIATION EFFECTS	73	589	75	41
085	ROCKET MOTORS	29	1,200	79	42
170	GUIDANCE	32	1,599	72	43
292	RELIABILITY	57	1,208	78	44
187	PROPAGATION	53	798	81	45
099	SOLID ROCKET PROPELLANTS	62	1,090	65	46
178	SCATTERING	61	683	73	47
005	MODEL TESTS	70	797	63	48
147	STATISTICAL ANALYSIS	38	557	82	49
006	VIBRATION	52	778	87	50
006	CRYSTALS	74	690	65	51
273	TRANSISTORS	67	825	64	52
071	BIBLIOGRAPHY	42	438	88	53
062	STORAGE	102	536	65	54
006	SUPERSONICS	49	812	55	55
079	ELECTRONIC EQUIPMENT	50	898	81	56
082	ELECTRON TUBES	46	722	63	57
009	AIRCRAFT	41	882	90	58
247	SEMICONDUCTORS	101	777	63	59
100	FUZES	66	496	57	60
196	MILITARY REQUIREMENTS	54	492	67	61
148	VELOCITY	78	747	86	62
060	DIGITAL COMPUTERS	85	604	80	63
131	OXIDES	84	495	[illegible]	64
253	SATELLITE VEHICLES	157	725	74	65
099	JET FIGHTERS	21	309	65	66
076	ELECTRICAL PROPERTIES	93	590	71	67
292	SENSITIVITY	99	538	72	68
292	ERRORS	64	675	71	69
006	LAUNCHINGS	69	856	67	70
090	VULNERABILITY	58	700	77	71
060	COMPUTERS	72	645	76	72
104	OPERATION	59	638	77	73
160	METALS	44	502	63	74
247	DEFORMATION	71	523	53	75
187	SPECTROGRAPHIC ANALYSIS	75	362	6	76
200	PATHOLOGY	120	165	33	77
117	GASES	110	551	67	78
117	THERMODYNAMICS	81	394	78	79
108	CANADA	125	415	82	80
060	MAINTENANCE	103	576	64	81
227	SEARCH RADAR	45	898	59	82
187	ELECTROMAGNETIC WAVES	122	601	68	83
016	STEEL	60	521	67	84
006	SHOCK WAVES	21	583	71	85
025	ANTENNAS	62	614	69	86
099	REDUCTION	87	466	78	87
048	CHEMICAL WARFARE AGENTS	64	410	52	88
027	X-BAND	189	737	58	89
256	SHEETS	117	539	51	90
059	ATMOSPHERE	124	524	61	91
212	PLASTICS	80	506	67	92
187	ABSORPTION	151	434	70	93
187	REFLECTION	97	515	68	94
227	RADAR TRACKING	96	877	64	95
056	COATINGS	79	469	62	96
005	WIND TUNNEL MODELS	500	691	51	97
059	METEOROLOGICAL DATA	136	253	46	98
006	DRAG	105	603	48	99
049	SILICON	220	615	48	100



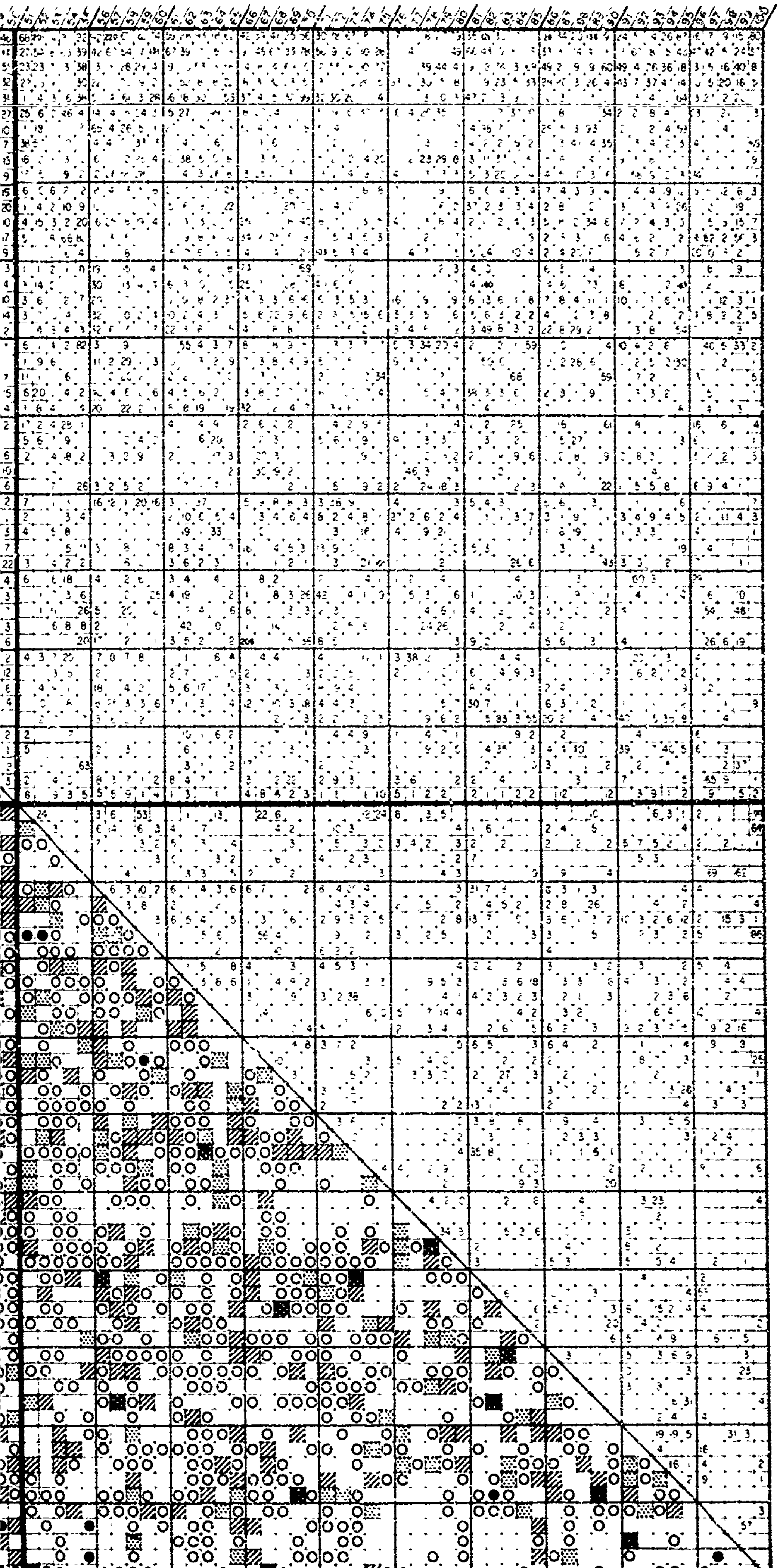


Chart 1

Pair Associations Among the 100 Most Common of the DDC Descriptors

(With Frequency of Occurrences)

Upper Right Triangle:
Actual Numbers of Occurrences

Lower Left Triangle:
Number of Occurrences by Ranges—

100 & over

10—24

50—99

5—9

25—49

1—4

None

Source of Data: Sample of 38,402 DDC Documents

e. Automatic Attribute Group Assignment of 599 Most Common DDC Descriptors. Consider first the initial establishment of the array of mutually exclusive attribute groups, which presupposes that a file of document descriptions exists. Without loss of generality, it can be postulated that all pair associations are formed prior to assigning any descriptor to a column in the array. This is equivalent to forming the record $D_{i,j}$ of associations for each descriptor and then beginning the assignment. This is the method followed in the computer program.

The program assigns the first descriptor, Rank 1, to attribute group (column) C_1 and sets up D_1, which at this point is the same as $D_{1,1}$. The next descriptor, Rank 2, is taken and D_1 examined to determine whether it contains the descriptor. If it does, the descriptor is not exclusive to C_1 and is placed in C_2, with D_2 being created.

In the general case, the next descriptor d_a is taken and each D_i, beginning with D_1, is examined to see whether or not it contains d_a. If it does--meaning that d_a forms a pair with one or more of the descriptors already in c_i--the D_i for the next column is examined. This process continues until one of two conditions terminates the cycle: (1) a column is found in which d_a does not appear in D_i, in which case d_a is added to that column (becoming $d_{i,j}$) and D_i is updated by logical superimposition of the descriptor's $D_{i,j}$; or (2) d_a is not exclusive to any column so far formed, in which case it becomes the first member $d_{m,1}$ of a new column C_m and its $D_{m,1}$ becomes the new D_m.

This technique assures that each descriptor is assigned to the leftmost (lowest-numbered) column c_i in which it is exclusive (i.e., is not used in c_i). The most time-consuming part of the computer operation is the handling of the tape containing the D_i records. Because each descriptor added to a column means superimposing its $D_{i,j}$ on the existing D_i, this tape must be rewound and rewritten for each descriptor processed. The tape containing the $D_{i,j}$ records is set up in rank number sequence and read only once during execution of the machine program. Allocating 599 descriptors required 3.75 hours on the UNIVAC II.

This algorithm assigned the 599 descriptors to 56 attribute groups, Table 1, each containing from 1-16 entries. It is not known whether or not this is a minimum. The technique assures that each descriptor is placed in the first possible column and, therefore, is used (forms a pair) with at least one other descriptor in every lower-numbered column. However, it is possible that movement of a descriptor from c_i into a higher-numbered column might eliminate some associations from D_i and thereby permit transferring other descriptors down into c_i. The most obvious candidate is the lone entry

in column 56; by juggling entries in some of the other columns, it might be fitted into one of the 55 left. If such a solution exists, it has not been found. The sheer volume of data involved precludes manual analysis and the thousands of possible rejugglings make a computer trial impracticable from a cost standpoint.

This 56-column array is almost twice as large as the 30 originally considered probable. Because the very frequently occurring descriptors form many different pairs--see Chart 1 and Table A-1 (Appendix 1)--and are rather broad in meaning, the need for assigning them into the attribute groups has been questioned. It may be more appropriate to make a separate list for each one of them and to restrict the lists formed by combining descriptor ranges in each of three columns to the attribute groups created from the less frequently used descriptors.

Choice of a cutoff point for this variation (which is not part of the original Multi-List System) is arbitrary. A new attribute group array was created after eliminating the 19 most common descriptors, all of which had 868 or more occurrences. The 580 assigned are all of the descriptors in from 72 to 846 document descriptions. The resultant array, Table 2, contains 46 columns. Although smaller than the first, it still is higher than the 30 thought possible.

The development of these two arrays is equivalent to their initial establishment and provides no information on the effects of using the same algorithm for a file-updating type of operation. Once initially established, the attribute group array requires updating (adjustment) as new documents add new descriptor pair associations to those already existing. Most of these involve descriptors in different columns and do not destroy the mutual exclusiveness within each. However, some new pairs involve descriptors in the same column and one of them must be moved, or renamed, to maintain exclusiveness. The UNIVAC II computer algorithm does not accomplish this renaming operation. Although the basic approach for modifying it to accomplish first-order renaming is relatively strightforward, actual computer time to run the program on even the fairly small set of 599 descriptors is excessive.

Nonetheless, it was considered pertinent to obtain some idea of the percentage of conflicts which result from adding new documents to an already-established file. For this purpose, the basic file was re-created after eliminating all pair associations occurring only in a random 10% sample--all document numbers ending in "9"--of the 38,402-document file. The algorithm then was rerun using the pairs remaining and the resulting attribute group array checked to determine how many conflicts would occur by introducing the associations found only in the 10% sample. (In effect, this considers the sample as constituting input to an updating cycle.)

The "9's" sample comprises 3,828 documents and includes just under 13,800 of the 49,306 different pair combinations in the full file. Of these, 2,062--about 15%--are unique to the "9's" documents; none occur in more than three documents. Occurrences in the full file and the sample are summarized:

DDC DOCUMENT FILE SAMPLE

Table 1
Mutually Exclusive Attribute Group Assignments of the 599 Most Common DDC Descriptors*

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	19	20	21	22	24	25	31	32	37	38	42
124	229	36	82	27	17	23	10	66	49	26	33	101	29	79	35	57	65	51	47	30	76	28	41	34	39	54	68
315	243	138	175	77	88	78	142	97	69	260	59	129	60	112	100	75	118	90	153	171	133	89	55	52	81	67	91
351	244	232	272	149	99	163	143	119	181	262	107	140	61	176	135	104	166	98	157	188	139	191	117	155	111	154	96
401	345	344	302	160	219	185	177	151	213	349	259	174	94	203	183	247	190	102	159	268	162	218	126	224	114	164	121
509	553	392	311	184	320	280	182	170	270	366	305	223	187	372	202	250	206	198	271	279	230	317	179	261	207	201	169
529	557	400	313	231	378	323	253	319	329	373	337	225	263	396	295	298	275	325	333	301	239	447	252	360	208	251	234
546	568	436	439	289	503	365	255	327	387	435	405	254	407	440	304	361	355	334	350	353	273	474	266	432	209	422	363
551		494	505	356	563	402	324	433	409	506	406	293	430	465	318	368	369	339	383	376	343	487	411	472	241	458	444
589		508	524	420	570	415	375	461	412	559	423	340	464	512	326	380	379	410	403	455	381		454	489	328	521	501
		564	560	481		431	417	511	492	567	438	414	503	533	399	451	382	471	449	483	390		507	491	357	565	514
		569	581	519		452	482	526	515	574	462	421	587	566	510	463	393	537	467	545	427			493	530	572	541
		594		584		480		591			520			579		466	561		468	548	457			593	544		549
		595				590		592			523					547			527		543			596			599
											552					577			562								
																			571								

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
43	44	46	50	53	56	58	62	71	72	73	80	87	106	113	115	120	145	178	199	200	204	216	257	294	321	371	441
45	64	48	63	85	70	109	86	74	108	83	95	116	110	131	125	137	173	194	222	240	228	284	287	332	388	496	
93	92	105	134	103	84	123	147	141	127	122	144	237	130	189	136	161	196	220	227	297	276	300	312	419	485	534	
150	128	193	152	148	146	158	245	264	172	156	211	316	197	212	214	221	248	226	292	395	308	358	314	425	550		
269	132	210	168	205	180	238	265	267	186	165	296	331	215	233	235	236	352	242	354	397	342	362	322	445			
274	256	217	192	283	195	246	291	278	258	167	336	413	303	299	282	277	437	428	374	424	446	391	398	495			
286	477	249	386	539	348	281	370	330	285	288	347	500	338	306	389	307	460	497	488	532		450	431	525			
518	484	346	404		442	309	443	359	290	310	426	555	341	416	459	394	469	498	556	578			448	597			
575	573	377	456		470	367	501	475	385	335	429		384	418	598	516	531	502					517				
	585	473	486		536	478	580	538	554	364	476		408	453		576	535	528									
	588	513	499			479		542		558	490			522				586									
			582			540																					

*Descriptors denoted by frequency-of-usage rank number.

DDC DOCUMENT FILE SAMPLE

Table 2
Mutually Exclusive Attribute Group Assignment of the 20th-599th Most Common DDC Descriptors*

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
20	21	22	24	25	27	28	29	32	33	38	39	41	49	50	53	56	58	59	62	71	72	73
23	47	30	36	26	34	40	31	42	37	51	43	44	64	63	66	70	68	61	67	74	84	78
98	57	112	91	77	35	52	45	106	46	54	48	90	65	89	130	75	150	69	82	93	86	156
167	104	188	181	94	115	124	55	119	109	81	60	96	140	117	138	76	169	139	108	173	132	165
177	153	196	184	142	151	203	111	208	210	286	100	129	162	134	143	102	193	160	154	178	135	180
243	194	305	213	253	185	217	122	220	224	289	118	212	230	168	269	244	223	163	191	190	146	263
248	219	349	229	255	200	291	152	256	298	297	196	251	359	285	344	273	252	250	238	246	301	341
325	239	369	290	293	231	299	158	275	323	314	202	281	411	376	348	296	271	338	247	261	386	418
329	260	405	351	317	232	311	206	282	370	318	254	313	479	390	364	404	335	356	266	381	410	428
339	295	482	361	379	302	326	225	310	406	340	322	367	536	454	444	439	396	380	366	445	474	484
385	304	529	402	422	423	360	508	346	436	401	350	420	542	486	538	478	442	407	471	452	568	512
400	324	552	579	445	434	415	526	430	462	427	481	448	561	511		523	475	447	580	463	576	540
432	334	555		446	510	421	550	438	566	507		480	588	515			541	456		489		554
435	467	557		453	518	466	595	465	577	537		596		551			549	493		585		593
473	500	559		487	547	564		490		556				575			596	573				
494	528	563		505	567	571		501		562				583				587				
495		572			594	590		503		570												
509		591																				
545																						
584																						

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
79	80	85	87	105	110	113	121	125	133	137	155	189	199	233	237	249	277	300	321	374	391	441
83	92	97	88	120	127	123	126	144	159	183	170	197	222	240	257	270	294	306	331	398	416	531
99	95	101	116	136	128	131	145	166	171	215	204	201	227	241	287	276	307	328	332	419	472	569
107	114	103	147	157	148	141	149	172	175	216	226	205	274	264	308	280	409	343	353	431	476	
182	207	214	176	218	164	174	161	187	195	228	250	245	303	267	312	283	429	365	412	517	499	
209	292	259	192	315	186	221	179	211	235	236	278	265	309	345	316	347	450	389	413	560	535	
242	327	320	268	319	363	234	342	272	262	458	330	337	354	403	333	362	516	497	488			
383	368	457	336	355	373	279	377	284	288	504	371	393	468	433	375	372	524	553	532			
384	397	464	449	387	395	352	408	378	358	514	496	461	469	459	392	477	539					
388	437	534	451	394	426	357	424	425	455	519	506	522	502	485	399	492						
417	491	597	483	414	460	382	520		527	525	586		530		470							
	578	599	498	440	543	556	521			533			565		574							
			513	582	544								592		589							
			546																			
			548																			
			581																			

*Descriptors denoted by frequency-of-usage rank number.

No. of Pair Occurrences	Different Pairs in Full File	Different Pairs Found Only in "9's"
1	20,218	2,001
2	8,654	59
3	4,854	2

Using the 47,244 pair combinations in the 90% of the file, the algorithm assigned the 599 descriptors into 55 mutually exclusive attribute groups, Table 3, each with from 2 to 16 descriptors. This is one less than the 56 columns for the full file, Table 1. However, the dispersion of descriptors in Table 2 is markedly different. The first 28 columns have significantly more descriptors--369 against 351. Seven groups, compared with four in Table 1, have six or fewer descriptors each and eleven have either 15 or 16, against only three. The assignment is the same for the first 63 descriptors, differences beginning with rank number 64, but only Groups 1 and 3 in the final arrays are identical. However, although the two arrays are quite dissimilar for a difference of less than 5% in the number of pair associations included (47,244 and 49,306), it appears impossible to draw any meaningful conclusions from this fact. The basic files in both cases are large--34,500 documents or more--and the form of the final arrays is more apt to depend upon chance variations in the particular pairs present than upon some meaningful factor.

The 2,062 pair combinations unique to the "9's" sample create 60 conflicts with the descriptor assignments of Table 2; the first is the pair 2-174 in Column 2. The conflicts comprise about 3% of the new pairs and occur at the rate of one in about 65 new documents. Two new documents, in turn, generate slightly more than one new pair among the 599 most common descriptors. (It should be noted that some documents--possibly as many as 25%--do not have two descriptors among these 599 and create no pair entering into the algorithm.)

All of these conflicts must be resolved by the updating algorithm. In an attempt to ascertain some of the results of this renaming operation, the adjustments have been traced through partially on a manual basis.

The simplest renaming is first order--moving one of the two conflicting descriptors into a column in which it is exclusive. This must be done in some prescribed order--e.g., by columns from low to high, in ascending sequence on descriptor code number of the pairs, in sequence on their rank numbers, etc. Results differ depending upon the order of resolution and also upon how much of available knowledge is used at the time a particular conflict is resolved. If, for example, all new pair associations are posted before renaming begins, then only the 60 conflicts must be clarified and the new assignment is final for the cycle. On the other hand, if new associations are accessed in the prescribed order and conflicts resolved as they occur, then a renaming subsequently may give rise to another conflict above the 60 already known. Both methods have been carried out far enough to demonstrate that the resultant array has at least 57 columns. It possibly has more, because this point has been reached before half of the known conflicts have been resolved. From Table 1, it is known that a 56-column array is possible.

DDC DOCUMENT FILE SAMPLE

Table 3.

Mutually Exclusive Attribute Group Assignments of a 90% Sample of the 599 Most Common DDC Descriptors

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	19	20	21	22	24	25	31	32	37	38	42
124	174	36	[illegible]	27	17	23	40	66	49	26	33	101	29	51	35	57	55	64	47	30	76	28	41	34	39	54	68
315	229	133	[illegible]	77	88	78	75	97	69	260	59	129	60	132	98	104	117	125	112	171	133	89	85	52	81	67	91
351	244	232	[illegible]	100	99	163	172	119	181	262	90	140	61	186	135	154	118	145	153	188	139	191	126	102	111	152	96
401	345	344	[illegible]	184	219	185	182	151	213	349	107	223	94	190	149	160	128	167	157	268	162	210	193	233	114	164	121
509	553	392	302	207	320	280	249	170	231	365	269	225	176	203	183	239	155	248	159	279	230	317	322	274	208	251	198
529		400	313	254	378	311	253	290	329	366	305	271	263	310	259	247	169	277	324	301	273	447	394	360	209	284	202
546		436	361	289	503	323	255	319	387	408	337	293	395	465	265	292	206	314	333	325	298	474	411	388	222	413	242
551		494	375	295	563	350	404	327	409	417	423	335	444	491	386	304	275	334	389	376	381	480	426	430	328	422	348
589		508	439	336	570	402	452	353	412	435	438	340	464	550	405	368	326	343	403	424	457	487	[illegible]	466	357	445	
		564	505	356		415	482	433	492	506	507	414	533	554	450	373	382	385	440	455	521		469	534	399	530	
		569	524	369		438	488	461	515	567	520	421	561		513	380	418	410	467	483	531		477	596	443		
		594	560	481		519	526	511	523		558	475	583		537	451	437	427	512	545	536		554		463		
		595	572	484		557	547	591			562	478	587			514	549	471	566		543		574		493		
			581	544		590	568	592			584		599				577	538	571						552		
																	598	573	580								

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
43	44	46	50	53	56	58	71	72	73	80	103	108	113	116	120	137	144	199	200	204	267	281	297	307	321	374	
45	62	48	63	87	70	83	74	92	79	95	206	110	123	147	136	143	161	227	226	257	287	358	441	528	425	419	
93	156	105	65	165	84	109	86	172	115	122	127	130	158	189	215	187	178	261	238	312	342	371	456	565		473	
150	245	210	134	195	146	142	141	236	235	166	148	131	173	205	216	221	201	282	240	331	384	391	502			556	
224	246	217	168	306	180	179	278	252	241	220	194	212	197	316	228	237	211	300	352	431	390	470	535				
285	288	291	196	309	192	266	330	308	372	234	363	214	281	332	294	270	264	302	355	578	398	522	597				
341	367	370	283	346	428	310	359	338	459	250	397	256	458	354	397	303	429	379	472		446						
452	516	407	364	416	432	377	383	393	539	258	406	276	476	460	499	454	504	396	500								
462		468	486	449	497	496		485	540	299	517	296	495		501	555	548	420	532								
489		510		479	559			490	593	448	542	389					576		535								
541		518		498				527																			
575		525		582																							
				586																							

*Descriptors denoted by frequency-of-usage rank number.

Consequently, it again is concluded that <u>an updating algorithm limited to first-order renaming does not maintain a minimum array of attribute groups</u>.

Because of the large number of possible movements which must be examined, no attempt has been made to resolve conflicts with higher-order renaming algorithms.

f. <u>Summary of Manual Analyses of Multi-List System</u>. The nonmachine studies of necessity have been confined to a limited number of descriptors with a data volume small enough to permit human simulation of computer processes. Their purpose has been (1) to examine the effects of alternative choices of action in the renaming operation and (2) to determine the degree of complexity of renaming needed to maintain the minimum number of attribute groups. The results, reported upon in detail in [4]-[6], are summarized briefly here, but the attribute group arrays are included.

The first trial used the 50 most common descriptors and pairs among them with five or more occurrences. The associations are shown in Chart 2. By inspection, it can be seen that most of the first 19 descriptors (all except 12 and 16) are used with each other and therefore must be in separate columns. The remaining ones are placed initially in the first (lowest numbered) column to which it can be assigned based solely upon lack of association with the one descriptor at the head of the column. This initial assignment is shown at the top of Chart 3. It utilizes only the pair associations formed by the 17 descriptors in the first line of the array.

Descriptors 20-50 are then processed in sequence, each one adding the new associations formed by it with the remaining descriptors; e.g., processing descriptor (rank number) 20 adds its associations with 21-50, etc. Some of these newly introduced associations cause conflicts which require that one of the descriptors be moved into a new column. This is done using only associations so far known to determine exclusiveness and the assignment may be changed subsequently as the remaining descriptors are processed in turn. It will be noted that this process corresponds to a file-updating operation, with renaming limited to higher numbered columns.

There are four variations in the sequence of processing steps when a new pair $d_{a,b}d_{c,f}$ is introduced and causes a conflict (i.e., both are in the same column and one must be moved). Each variation results in a different final array:

Variation 1: Check $D_{a,b}$ before $d_{e,f}$. Move $d_{e,f}$ to a new group.

Variation 2: Check $d_{a,b}$ before $d_{e,f}$. Move $d_{a,b}$ to a new group.

Variation 3: Check $d_{e,f}$ before $d_{a,b}$. Move $d_{a,b}$ to a new group.

Variation 4: Check $d_{e,f}$ before $d_{a,b}$. Move $d_{e,f}$ to a new group.

*Descriptors denoted by frequency-of-usage rank number.

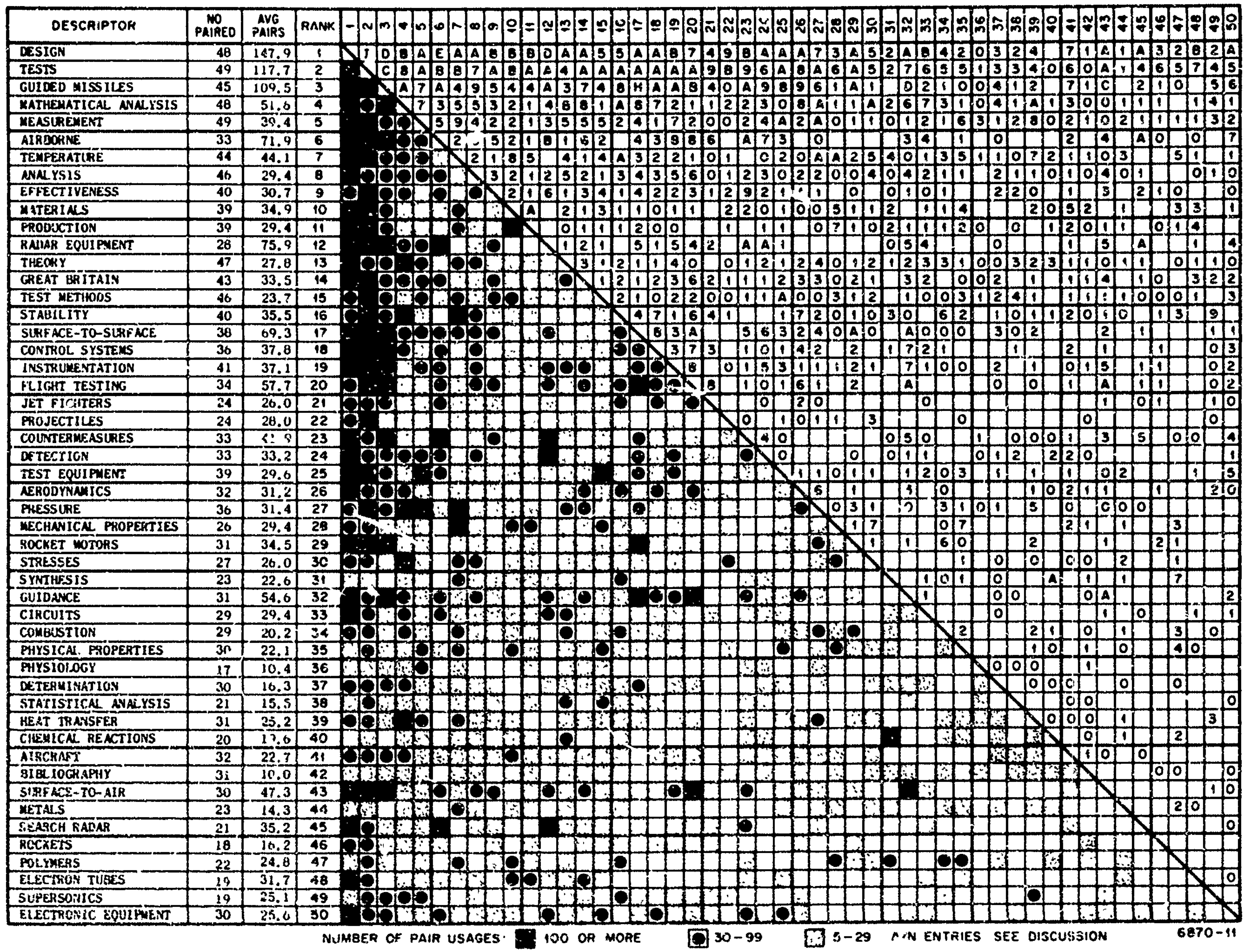


Chart 2

Pair Associations Occurring Five Times or More Among the 50 Most Frequently Used ASTIA Descriptors

Observe that the solutions differ only when a choice of actions is possible--that is, when both descriptors are exclusive to the first group in which either is exclusive, or when both are inclusive to all established groups. Once an initial difference of action has taken place, many of the subsequent renamings, of course, are different, although some are still identical. It is to be expected that the four solutions will all differ from each other.

Results of processing each of the four variations are shown in Chart 3, 22 or 23 groups being required for the 50 descriptors. No conclusions can be drawn as to which variation is preferable; the fact that two of them require one more column may be due only to the particular data in the experiment and cannot be used to conclude that they are of necessity less preferable. The different results do indicate that it may be extremely difficult to select a sequence of operations which will assure a minimum number of attribute groups.

Whether or not the number of columns (22) is actually minimum is unknown. It has been proved that, with these data, at least 21 are required. However, attempts to reduce the array to 21 columns have been unsuccessful and, similarly, no proof has been developed to show that 22 are necessary.

A series of manual simulations, identical in approach to the foregoing, then was performed on the 100 most common descriptors, using all pair associations existing among them. (The associations are shown in Chart 1.) Results are present in Chart 4, with from 39 to 42 columns being required, depending upon the variation chosen. The array of 4E is a further modification of the procedure creating 4B and is included to illustrate the effects of retracting renamings which subsequent actions show to be unnecessary. Thus, if $d_{a,b}d_{e,f}$ conflict and $d_{e,f}$ is moved, a pair association introduced at a later time may result also in moving $d_{a,b}$ into a new column. But this may make it possible to restore $d_{e,f}$ to the original column. This procedure was followed in creating the array of Chart 4E. The array of 4F follows the logic of setting up the attribute groups initially, using all pair associations in the data to guide the assignment.

The arrays generated with the sets of both 50 and 100 descriptors have been set up using first-order renaming only. All have been reviewed for reduction in number of columns through more complex renamings and, mostly by chance, it has been shown that the arrays 4B, 4C and 4D can be cut one column. That reducing 4C to 38 columns is most sophisticated, involving second-order, third-degree renaming. It is reduction that introduces the concept of nth-degree renaming to the nth-order renaming initially proposed for the Multi-List System.

The results of these simulations bring out several significant factors pertaining to the establishment and maintenance of descriptors in a minimum number of mutually exclusive attribute groups.

First, file storage requirements for holding descriptor pair associations are large. A record must be maintained for each descriptor showing every other descriptor with which it is used in a document description. In the case of the DDC sample, this auxiliary file is more than twice as large as the basic document/descriptor file itself. Furthermore, most pairs occur

Initial Assignment

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2	3	4	5	6	7	8	9	10	11	13	14	15	17	18	19						
40			47																			
		31			16	12		35	21	20				22			24					
		44			26	23			32	50				41			25					
		48			28	45			39								27					
					29	46			43								33					
					30																	
					34																	
					36																	
					38																	
					39																	
					42																	
					49																	

FINAL ARRAY: $d_{a,b}$ Compared First. $d_{e,f}$ Moved to New Group

1	2	3	4	5	6	7	8	9	10	11	13	14	15	17	18	19	20	24	25	29	43	
40		31	47		16	12		35	32	23	21	34		22	28		30	27	42	50		
		48			36	46			45	26		38		41	39		33					
						49				37				44								

FINAL ARRAY: $d_{a,b}$ Compared First. $d_{a,b}$ Moved to New Group

1	2	3	4	5	6	7	8	9	10	11	13	14	15	17	18	19	29	20	24	25	32	
40		31	47		16	12		35	43	23	21	34		22	28		50	30	27	42		
		48			36	46			45	26		38		41	39			33				
						49				37				44								

FINAL ARRAY: $d_{e,f}$ Compared First. $d_{a,b}$ Moved to New Group

1	2	3	4	5	6	7	8	9	10	11	13	14	15	17	18	19	39	23	24	25	26	43
40		44	47		16	12		28	32	20	21	34	31	22	30		50	29	27		33	
					36	46		48	45			38		41				37			35	
						49												42				

FINAL ARRAY: $d_{e,f}$ Compared First. $d_{e,f}$ Moved to New Group

1	2	3	4	5	6	7	8	9	10	11	13	14	15	17	18	19	23	24	25	29	39	50
40		31	47		16	12		28	32	20	21	34		22	30		26	27	42	33	43	
		48			36	46			45			38		41			35					
						49								44			37					

Chart 3

Final Arrays Resulting From Four Variations of First-Order Renaming
(50 Most Common DDC Descriptors, Pair Associations With 5 or More Occurrences)

Chart 4

Final Mutually Exclusive Attribute Group Assignments for the 100 Most Common DDC Descriptors (Five Variations)

Group	A: Initial Assignment	B: Final Array—$d_{a,b}$ Compared First. $d_{e,f}$ Moved to New Group	Descriptors Exclusive to Final Array	C: Final Array—$d_{a,b}$ Compared First. $d_{a,b}$ Moved to New Group.
1	01	01	N	01
2	02	02	N	02
3	03 36	03 36	N	03 36
4	04 81	04 81	N	04 81
5	05 27 74 76 77 88 100	05 74 77	N	05 88 100
6	06 17 63 95 99	06 17 88 99	N	06 17 99
7	07 23 29 33 35 62 75 78	07 27 62	N	07 62
8	08 40 48 65 86 97	08 86 97	N	08 86 97
9	09 66 96	09 66 96	N	09 66 96
10	10 15 49 60 70 91 98	10 15	N	10 70 91
11	11 26	11 26	N	11 26
12	12 59 64 68	12 33 68	N	12 78
13	13	13	96	13
14	14 61 94	14 29 60 61 94	N	14 29 60 61 94
15	16 41 67 84 85 90 93	16 41 90 100	88	16 67 90
16	18 57	18 57 75	77	18 57 75
17	19	19 65	N	19 65
18	20 51 79	20 79 98	100	20 79 98
19	21 45 47 82 89	21 47	57 61	21 45
20	22 30 46 50	22 50	N	22 50
21	24	24 91	36 77	24 76 77
22	25 28 32 54 83 92	25 32 89	N	25 32
23	31 55	30 31	77	31 33 55 84
24	34 52 53	34 53 85	N	34 53 85
25	37 39 42 56 72 87	37 63 64	[illegible]	37 72 93
26	38	38 95	29 36 54 64 74 76 77 84 96 98 100	27 38 89
27	43	23 43 48	47 60 77 88	23 43 47 48
28	44	44 46 55	29 77	39 44 92
29	58	51 54 58	48 85	52 54 58
30	69	28 69	60	28 69
31	71	39 59 71	29 77 91 98	30 71
32	73	45 73	23 29 33 50 77 99	59 73 83
33	80	35 80 82	60 77 100	35 80 95
34		40 42 67	75 88	15
35		49 52 92	34 47 55 64 85 99	40 42 51
36		56 76 84	55 69 77 97 98 99	41 87
37		70 78	23 60 61 75 90 94 96 98 100	46 56
38		72 93	37 70 74 75	49 74 82
39		83	31 DIFF	63 64 68
40		87	21 DIFF	
41				
42				

(N = None)

Page 1 of 2 pages

Chart 4

Descriptors Exclusive to Final Array

N	N	N	N	N	88	27 100	N	N	60 77	N	77 96 100	96	N	88	77	N	100	88	N	96 98	89	77	N	N	77 93 98	N	29 77 97	85	48 60	51 68 77 100	33 77 84 99	60 77 100	10 33 79 90 95	29 75 88	37 70 88 90 100	55 75 77 90 94 98	55 99	65			

D: Final Array—$d_{e,f}$ Compared First. $d_{a,b}$ Moved to New Group

01	02	03 36	04 81	05 74 77	06 17 88 99	07 27 62	08 48 97	09 66 96	10 70 91	11 26	12 33 68	13	14 29 60 61 94	16 41 90 100	18 57 75	19 65	20 35 98	21 86	22 50	24 76	25 28 89	31 79	34 53 85	37 87 93	38 54 59	23 43 47	44 46 55	51 58	69 78 92	30 71	45 73	80 82 84	15 95	32 52	39	40 42 67	49 64	56	63	72	83

Descriptors Exclusive to Final Array

N	N	N	N	N	N	N	N	N	60 77	N	N	96	N	88	77	N	100	88	N	77 96 98	N	29 99	N	N	29 94 98	48	29 77 94	48 54 85	77 97 100	51 68 100	23 29 33 50 77 99	33 48 60 97	33 90	34	28 D I F F	75 88	34 52 55 65 70 82 86 89 92 99	18 D I F F	19 D I F F	22 D I F F	31 D I F F

E: Final Array—$d_{a,b}$ Compared First. $d_{e,f}$ Moved to New Group. Unnecessary Renamings Retracted.

01	02	03 36	04 81	05 27 77	06 17 88 99	07 62 100	08 86 97	09 66 96	10 15	11 26	12 33 68	13	14 29 60 61 94	16 41 90	18 57 75	19 91	20 79 98	21 47	22 50	24 76	25 32 89	30 31	34 45	37 56 93	38 74 95	23 43 85	44 46 55	52 54 58	28 48 69	39 51 71	59 73 83 84	35 80 82	40 42 67	49 64 70	53 87	63 65	72 92	78			

F: Final Array—"One-Shot" Initial Assignment Method

01	02	03 36	04 81	05 27 77	06 17 88 99	07 23 48	08 40	09 66 96	10 15	11 26	12 33 59	13	14 29 60 61 94	16 35 100	18 57 75	19 65	20 51 90 98	21 45	22 30	24 76	25 28 89	31 41 55	32 34 52	37 39 82 91	38 54 67	42 68 97	43 47 48	44 46	49 64 70	50 63	53 85	56 69 84	58	62 86	71 74 93	72 92	73 79 83	80 95	87		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42

Page 2 of 2 pages

only once or twice and new pairs are added at a fairly high rate as new documents are introduced. Other libraries may have characteristics different from that of DDC and, presumably, at some stage of library size, the number of new associations becomes rather small. It is not known at what point this occurs.

Second, if both of two conflicting descriptors meet the algorithm specifications for renaming, then the one selected apparently has an effect on the number of groups. Results of the analyses do not indicate which, if any, of them maintains the minimum number of columns. Indeed, it is possible that each of the several variations should be followed through to completion to determine which is best. Even a few conflicts gives a large number of possible combinations of actions.

Third, a fairly complex renaming process is necessary to achieve a minimum array. It has been shown that first-order renaming does not do this. The analysis indicates that renamings of higher order, and also of higher degrees, are required, but how high has not been established. In any case, the number of possible actions to be investigated rapidly becomes so large as to pose an almost prohibitive processing workload. Even second-order renaming--the simplest after first order--involves, with N descriptors, looking at close to $N(N-1)$ possibilities. It appears, but has not been proved, that even more complex renaming is needed to maintain a minimum array.

Finally, resolution of conflicts using the algorithm of Figure 1 does not necessarily maintain a minimum number of groups. In this algorithm, a renaming process is initiated only if a new association occurs for two descriptors, previously mutually exclusive, in the same column, and one of them at least is involved in the resultant renaming. It is perfectly possible, however (see Chart 4E), to reduce the array after the updating cycle by movement of one or more descriptors for which no new associations have been introduced; indeed, there may be no new pairs in that column. It may well be that the complete array should be reanalyzed after each updating cycle to be certain that it is the minimum achievable. Otherwise, it may continue to expand over a period of time until it contains several more than the minimum number of columns.

No attempt has been made toward determining the approximate complexity of renaming required for a minimum number of attribute groups. The array already is so large that a reduction of even 3-4 columns does not negate the conclusions of the next subsection.

g. <u>Evaluation of the Multi-List System in an IS&R Application</u>. Evaluating the potential usefulness of the Multi-List System in a document retrieval application reduces essentially to answering one question: "Is it practicable to use combinations of several mutually exclusive index terms as superkeys identifying lists into which the file is organized?" Based upon the studies and simulations which have been completed on the DDC data, the answer must be an unqualified "No." As a corollary to substantiating this conclusion, it is possible also to establish the general data characteristics which file records must possess if the approach is to have potential merit.

The notation $S_{i,j} = d_{r,a}d_{s,b}d_{t,c}$ is used for a superkey, where c_r, c_s and c_t can be taken as three consecutive attribute columns and the $d_{r,a}$, $d_{s,b}$ and $d_{t,c}$ are the groups of index terms in each column combined into the superkey range. For simplicity, the discussion assumes that each column can contain about the same average number of terms without increasing the number of columns in the array. Whether or not the assumption is true does not affect the analysis.

The pair associations of the 599 most common index terms in the DDC array require an array of 56 exclusive groups; including all 5540 terms in the sample may increase this to about 60, or 20 sets of three columns. The documents in the sample average only 5.14 index terms each; in the latter part of the chronological period represented, this increases to between 7.5 - 9.1, depending upon security classification. In the entire sample, only 859 documents have 12 or more terms, the maximum being 21. The attribute group array, then, has several times as many columns as documents have descriptors. In fact, for the 97.5% of documents with 11 or fewer describing terms, there are at least five times as many columns.

Although the distribution of index terms among columns is not random, it is evident that any one document has terms in only a few of them. In fact, <u>if it has an index term in the range $d_{r,a}$ in c_r, there is a high probability that it has none at all in c_s and c_t</u>. Yet this one term must be included in a superkey $S_{i,j}$ which, by definition, includes a range of terms from each of the three columns. Fulfillment of this requirement leads to the concept of the null index term, Δ_i, present in every column c_i and defined as being exclusive to all real terms. In this example, the superkey then becomes $d_{r,a}\Delta_s\Delta_t$ and it is quite evident that most superkeys will be of this type--probably well over 90% of them. Some will have two real terms, with $S_{i,j}$ expressible in the form $d_{r,a}d_{s,b}\Delta_t$, and only a small percentage will have a term from each of the three columns. For practical purposes, then the superkeys for most of the index terms in a document will contain one real and two null terms and be of the form $S_{i,j} = d_{r,a}\Delta_s\Delta_t$. This is equivalent to establishing a separate list for each index term (or small range of terms), except that additional bits and file storage space are needed to denote the nulls.

As a direct corollary, the lists containing only a single real index term or range, although relatively few in number (600 if the 60-column array is broken into ten ranges per column), contain most of the entries criss-crossing the file storage area, but each list is long. Lists entered by superkeys of the form $S_{i,j} = d_{r,a}d_{s,b}\Delta_t$ or $d_{r,a}d_{s,b}d_{t,c}$ may be many more in number, but each is short.

Because index term assignment to columns is not random, it is possible that some "clustering" tendency may exist in actual usage, especially if the mutually exclusive columns are reordered. To investigate this possibility, the high-frequency index terms in 50 consecutive documents in the "9's"

sample were listed by their rank number and column assignment in the array of Table 1. (Because only those with two or more high-frequency descriptors are involved, the selection range covered 61 documents.) These are listed in Table 4, both rank numbers and attribute group columns being given in ascending order; i.e., there is no column correspondence between the two. Inspection of the right half of this table clearly shows that almost all superkeys are of the form $d_{r,a}\Delta_s\Delta_t$. All attempts to improve results by reordering columns have been fruitless. Because it does not include the less frequently-used index terms, this table not fully definitive. However, the dispersion of columns is so great that even their optimum placement for each document--an impossible expectance--would not significantly alter the table.

Processing search requests against a file list-organized in this manner introduces a similar set of considerations. Consider a subset of the search request in which terms are connected by a logical "and" relationship. In most cases, each single term will form a super key of the type $d_{r,a}\Delta_s\Delta_t$. But it is not sufficient to search only the list entered by this superkey; <u>all</u> lists of which $d_{r,a}$ is one member must be searched. If each column is divided into ten ranges, this is 100 lists, of which some may have no entries. As noted above, many of these are short, but nonetheless the records they include must be examined.

It is concluded, therefore, that the grouping of index terms into mutually exclusive attribute columns and the organization of the file into lists entered by multi-column superkeys has no potential usefulness in a document retrieval application and is, in fact, markedly inferior to a straightforward list-organized file in which each term has its own list. Exhaustive analyses of the large DDC sample show that the typical superkey contains only a single real index term (or range of terms). The Multi-List System does not achieve its objective of minimizing search time. Compared to a conventional list-organized file, the Multi-List System organization not only requires searching many more lists in processing retrieval requests, but also uses considerably more data storage space for retention of the pair-association lists. Finally, maintenance of the attribute group array imposes an additional processing workload conservatively estimated to be two or more orders of magnitude greater than that needed for single-term lists.

It may be argued that the characteristics of the DDC file are not representative of those in other document retrieval applications and that the above conclusion is not generally true. To some extent, this may be valid. We are unaware of any published results of analyses into file characterists which have been carried out on the scope or into the depth of those on the DDC sample. Those data which have been noted are considered to be consistent with these results. In fact, there is some reason to believe that more-specialized document files, indexed in greater depth and with a smaller thesaurus than the DDC library, may show an even more unfavorable relationship between number of index terms per document and the size of the mutually exclusive attribute group array.

DDC DOCUMENT FILE SAMPLE

Table 4.

High-Usage Descriptors and Their Mutually Exclusive Attribute Group Assignments for a Selected Range of 50 Documents

Doc. Number	No. Descr.	Rank Numbers of High-Usage Descriptors*	Mutually-Exclusive Attribute Groups (Table 1)
230019	6	044 305	12 30
230029	3	174 304	13 16
230049	9	003 009 096 178 183 185 213 359	3 7 9 10 16 28 37 47
230059	9	003 004 090 337 352	3 4 12 19 46
230069	5	417 526 562	8 9 20
230079	10	083 169 176 247 256 422 445	15 17 27 28 30 39 53
230089	6	049 076 595	3 10 22
230099	5	011 416	11 43
230119	12	014 036 118 318 536	3 14 16 18 34
230139	9	001 031 131 164 314 455 498	1 21 24 27 43 47 52
230149	7	003 046 078 083 252 332	3 7 24 31 39 53
230159	8	003 009 031 169 256 358 369	3 9 18 24 28 30 51
230179	5	064 222 384	30 42 48
230199	8	008 022 386	8 21 32
230219	5	030 143 167 237	8 21 39 41
230229	3	030 167 578	21 39 49
230239	12	001 007 017 031 052 059 095 182 528	1 6 7 8 12 24 25 40 47
230259	5	003 150	3 29
230269	8	004 018 094 438	4 12 14 17
230289	8	037 152 220 240 295 344	3 16 26 32 47 49
230309	6	041 177 192 223	8 13 24 32
230319	8	002 115 233 281 292 493	2 25 35 43 44 48
230329	8	035 050 189 207 358 459 502	16 26 32 43 44 47 51
230339	4	002 093	2 29
230349	7	311 339	4 19
230359	5	001 057 187 592	1 9 14 17
230379	10	020 021 147 222 490 496	19 20 36 40 48 55
230389	5	001 010 087 447	1 10 23 41
230419	7	009 065 200 391	9 18 49 51
230439	4	003 009 167	3 9 39
230459	9	001 004 144 147 249 262 352 544	1 4 11 26 31 36 40 46
230469	9	001 029 091 127 413	1 14 28 38 41
230509	3	173 184 426	5 40 46
230519	10	002 007 105 289	2 5 7 31
230539	6	001 018 063 237	1 17 32 41
230549	7	029 232 511	3 9 14
230579	8	002 004 029 259	2 4 12 14
230589	5	047 158 195 430 509	1 14 20 34 35
230609	5	001 002	1 2
230619	15	001 036 039 316 355 429 431 486	1 3 18 26 32 40 41 52
230639	10	175 269 438 576	4 29 45 48
230659	6	009 141 195	9 34 37
230689	11	003 004 009 046 094 192 203 247 300	3 4 9 14 15 17 31 32 51
230699	7	076 208 560	4 22 26
230729	10	011 076 236	11 24 45
230749	6	218 404 460 493 546 578	1 23 25 32 46 49
230769	7	027 051 119 193	5 9 19 31
230789	3	052 100 361	16 17 25
230799	6	077 200	5 49
230809	6	033 074 449 501	12 15 20 36

*Includes only documents with two or more high-usage descriptors. 11 documents in this range have none or one.

h. <u>Characteristics of a File Suitable for Multi-List System Organization</u>. The evaluation implies one basic characteristic a file must possess if the Multi-List System is to have potential usefulness: There must be a number of data fields having a range or set of different values as possible entries and in which all, or nearly all, file records do have an entry. These need not comprise all the data fields in the record; it can be divided into parts, one consisting of fields present in practically all records and the other, the variable or trailer fields which may or may not exist in every record. The first type can be combined into superkeys; the second, included in individual lists.

Many types of files have records with this characteristic. In general, the fixed fields may be further subdivided into two categories. First are those in which a single record can have only one entry, such as clock number, base hourly wage rate, home department, number of dependents, etc., in a personal file. Each such field may be considered as an attribute, for which the possible values are by definition mutually exclusive and no procedure is required to maintain this exclusiveness. Second are those fields in which a single record may have multiple entries, such as foreign language proficiency and higher job categories for which a person is qualified. Here exclusiveness is not an a priori condition, but is a function of the particular entries which exist in the totality of file records. If such attributes are to be divided into several (two or more) mutually exclusive groups, then the file maintenance procedure must provide for retaining a record of existing associations and adjusting the groups as necessary to reflect changes in the detail entries.

Attributes with only a single possible entry per record probably are most susceptible to grouping into superkeys when a list-type file organization is being evaluated. By attaching a chain address to groups of two or three data fields (attributes), range broken into superkeys, rather than to each one, some storage space and input-output transfer time always can be saved in handling an individual record. Additional savings may accrue by using condensed codes for the superkeys themselves. Such savings, however, may be only a fairly small percentage of the record size in a normal list-organized file.

At least partially offsetting this gain is, usually, somewhat increased complexity and, possibly, additional computer time both in maintaining the file and in processing search requests into the lists. This arises because a search may not--and normally will not--involve all of the attributes grouped into a superkey, but will be of the form $d_{r,a}\lambda_s\lambda_t$ or $d_{r,a}d_{s,b}\lambda_t$, for which several lists must be searched. Although exactly the same number of records may be examined with either single-value or superkey list organization, the latter method requires the extra machine instructions and computer time for accessing, transferring and examining records in many lists instead of just one. Evaluation of the relative payoffs, and of the efficiency of organizing the file itself into lists, must be predicated upon the total uses of the file and cannot be done here.

If the mutually exclusive attribute groups have to be developed and maintained in the manner necessary for a document retrieval application, then it is concluded that the Multi-List System does not constitute a feasible

method of file organization and use. Standard approaches are superior in terms of storage requirements, of file maintenance complexity and processing time and of file searching and use. It may result in reduced storage requirements for files with many data fields of the "attribute" type, but in most cases will require more processing time for file updating and use.

C. ANALYSIS OF THE BLACK-PATRICK VARIATION OF A DOCUMENT-SEQUENCED FILE

D. V. Black and R. L. Patrick have suggested [9] a variation in the document-sequence file as a means of realizing greater file-searching efficiency. In this approach, the index terms for each document are ordered in ascending sequence (as one possibility) on their code numbers and the file records, document numbers and index term codes, are ordered on the string of code numbers considered as a single variable-length key. Where keys are identical, records are in document-number sequence. A file so organized looks like this:

Doc. No.	Term 1	Term 2	Term 3	Term 4
1000	123			
9000	123	234		
7000	123	234	345	
4000	123	234	345	567
4001	123	234	345	567
4002	123	234	345	567
3053	742	999		
0123	846	978	1235	
8421	847	1341		
9766	954			

It will be observed that the records are identical to those in the normal document-sequenced file, in which index terms usually are carried in ascending sequence on code number within each document. Only the record sequence within the file is different.

The index terms in a request (assumed here to have logical "and" connectives) are converted to code numbers and similarly ordered into ascending sequence. In processing the request, searching need continue only through that portion of the main file in which the first terms are equal to or less than the first term of the request. For example, if a search includes the terms 234-345-567, the search through the "file" in the table

above terminates after document number 4002. Because the documents beginning with 3053 include no index term less than (code number) 742, they obviously cannot meet the search criteria. In the portion of the file in which a "hit" is possible, each file record is examined by a conventional comparison subroutine to determine whether or not it meets the criteria.

Does this approach have any significant potential in a document retrieval application? Unquestionably, it permits terminating a search without examining all the documents in the file and, from this standpoint, is preferable to a straight document-sequenced organization. The percentage of the file records that can be bypassed, on the average, has not been reported. In fact, so far as known, the proposal has not been tested against an actual file of document descriptions and a representative sample of search requests.

If documents are ordered on the lowest index term code in their descriptions, there is obvious tendency for the file records to be clustered among the lower code numbers. Further, the probability of having a low code in a description increases with the number of terms used. Both of these tendencies are evident in this summary of 50 DDC documents classified by low descriptor code used. (These are document numbers ending in "9" in the DDC accession number range 229009-229499, described in 1960. It is not a random sample but is considered roughly typical of documents accessed during that period.)

Low Descriptor Code Range	Number of Documents	Average Descr. per Document	Cum. % of Documents
0001-0199	9	8.67	18 %
0200-0399	8	7.75	34
0400-0599	5	8.40	44
0600-0799	-	-	44
0800-0999	5	9.20	54
1000-1199	-	-	54
1200-1399	8	5.75	70
1400-1599	1	3.00	72
1600-1799	2	5.00	76
1800-1999	1	5.00	78
2000 & Up	11	4.91	100
Total	50	6.92	-

Only two documents have lowest codes greater than 3000--3204 (three descriptors) and 4779 (four descriptors). Thus almost all these documents have at least one index term in the first 40% of the descriptor code range (maximum about 7000) and over half of them are in the lowest one-seventh (below 1000). Because DDC codes are assigned sequentially to descriptor names in alphabetic order, this clustering tendency in the lower number range is equivalent to saying that most documents are described with a term whose first letter is early in the alphabet.

The sequenced codes for the terms in a set of average search requests likewise have a clustering tendency, not necessarily the same as that exhibited by the library as a whole. The portion of the file that can be bypassed in processing them cannot be estimated with any accuracy without conducting an analysis using descriptions of a reasonably large collection of documents (several thousand, at least) and a representative cross-section of search requests.

The number of documents examined might approximate, for example, half the library if the average request meets four conditions: (1) The number of terms is fairly small; (2) terms have only logical "and" connectives; (3) retrieval is based upon a full match of all terms and not varying subsets of those in the request; (4) the average request is described to about the same degree of detail as the average document; and (5) over a period of time, the distribution of subject classifications in search requests approximates that of the document library. In practice, these conditions are not met and the general effect of the deviations is to increase the portion of the file which must be searched.

In the conventional document-sequenced file, new documents can be added at the end with insertions (if any) limited to the latter portions of the file. In the Black-Patrick variation, insertions are the rule and the entire file must ' rewritten on each updating cycle. To this extent it imposes an additional processing workload and cost. Although no experimental data have been seen to support the conclusion, it appears quite possible that the method is preferable to the standard document-sequence file, where a saving of even 10% in the number of records examined may be profitable. However, it is not considered competitive with either the inverted sequence or list-organized file in processing search requests. It is applicable only with magnetic tape or other sequential-access storage medium and, despite the fact that a list-organized file is twice as large, the latter almost invaribly will result in lower over-all processing time and cost.

D. OPTIMUM ORGANIZATION OF A DOCUMENT RETRIEVAL FILE

There seems to be rather general, but not universal, agreement that, for the foreseeable future, automated document retrieval will based upon searching a file in which documents are described by index terms and in which the request terms are connected by varying complexities of logical "and," "or" and "but not" relationships. There also appears to exist rather general concurrence--possibly not quite so pronounced--that only the inverted sequence and list-organized files provide really efficient means for automated retrieval. Certainly only these two can be considered in a real-time operation, which demands an on-line, mass-storage (random access) device for the document file.

1. General Comments on Factors Affecting File Organization.

The most efficient detailed form of file organization is predicated to some extent upon characteristics of the data processor and its storage devices. For example, if a disc file or drum always transfers blocks of 100 characters, nothing can be done about it (without changing the equipment) and the detailed file design and use specifications must take this fact of life into account. Insofar as internal processing and data storage capabilities are concerned, practically all modern (current decade) general-purpose EDPM's are quite flexible and pose no basic restrictions on the type of file organization established. A real-time retrieval operation--and particularly one in which a person is permitted to "browse" through the automated file--requires some type of query (data input) and display (data output) device connected to the processor. Here the limitations are much more apt to be those of the capabilities--and cost--of the device rather than those of the rest of the processing system. Because of these equipment-related factors, a detailed file layout can be made and optimized only within the framework of the characteristics of a specific equipment configuration.

The most efficient general form of file organization, however, depends largely upon the requirements the file processing must meet and the environment in which the operation is performed. Consequently, it can be studied and conclusions can be reached. This is true despite the fact that requirements and environments are quite diverse and, at first glance, it might seem that the optimum file organization takes many forms, depending upon the particular conditions applicable. The problem can be reduced to manageable size by eliminating those phases or requirements which are not a direct part of maintaining the index file to be searched or of processing requests against it.

As examples, the procedures for maintaining and using an automated thesaurus are essentially identical for both list-organized and inverted files. The method of arriving at index terms--manual or machine--and of validating them against the thesaurus is a function independent of the organization of the index file. The accumulation of statistical data can be done in about the same way with either type of file. A similarly separate processing function is that of maintaining auxiliary files which may be required, such as those used to develop significant usage associations of index terms. Selection of documents for "current awareness" programs occurs at the time new descriptions are entered for processing and also is independent of the particular file format in which the data are to be stored for subsequent searches.

2. Advantages and Disadvantages of Inverted and List-Organized Files.

The organization, content and use of the index term file are predicated upon the requirements of the search algorithm and the exact nature of the output. Both the inverted and list-organized files contain only document numbers and index terms. The output of a search through either type of file, then, is limited to these two types of data. Inclusion in the output of such additional information as titles, abstracts or copies of documents is not possible using only these files, but requires one or more additional operations. These are not part of the direct file searching process and may or may not be automated.

a. <u>Differences in Search Outputs</u>. The first basic difference in the use of these files is the nature of the output. For practical purposes, the output of searching an inverted file is a list of each document number satisfying the search criteria plus, if desired, the list of index terms upon which the selection was based. The list-organized file can produce not only the document list but also <u>all</u> index terms used in each description. In addition, by expanding the size of the file record, such other data as author's name, publication or journal, date of publication, etc., can be incorporated in the output. This is possible whether or not such fields are used in the same manner as "normal" index terms.

The greater output flexibility of the list-organized file points out another essential difference between the two types. The fact that it is based upon a document record which can be expanded rather easily to include more data than the basic indexing terms themselves is a strong incentive to do just that. Consequently, the evaluation of which type of file is most efficient usually will not be based upon two different organizations of the same data base. Almost inevitably, the list-organized file will contain more information than the inverted file.

If output requirements are satisfied by a list of document numbers (plus, at most, the descriptors upon which the selection is based), then either type of file organization can be used. If additional descriptive information of the general types mentioned above are postulated, then only the list-organized file is applicable.

b. <u>Differences in Nature of Search Algorithm</u>. The list-organized file is more flexible than the inverted file in the degree of sophistication or complexity permissible in the search algorithm. The list-organized file can be used for any type of search possible against an inverted file. In addition, it permits search criteria which are not practicable with the latter form of file organization. The greater capabilities of the list-organized file arise because, in processing a request, it makes available more data than does the inverted file.

The relative degrees of search complexity may be summarized in this manner: With an inverted file, all index terms used in the selection must be contained in the basic search request, or must be derivable from sources other than the file itself. As examples of the latter, the input terms may be expanded based upon hierarchal or structural relationships carried in an (automated) thesaurus, or upon usage association data contained in the thesaurus or other file which can be accessed with an index term as key. In addition to the above, the list-organized file makes it possible to incorporate criteria based upon terms contained in document records accessed through the initially given terms. The additional terms so obtained are derivable only from within the list-organized file itself.

The applicability of the two files to some of the more commonly proposed search parameters are discussed briefly:

Both can handle the same complexity of logical relationships between search terms; typically limited to "and," "or" and "but not" connectives.

Both have the same capabilities for converting between external and internal language: Term names to index term codes, non-indexing names or codes to indexing codes, external index codes to internal codes, etc.

Both files can handle requests when all terms in the search criteria are included in the request input.

Both files can be used when the selection criteria can include subsets of the full range of index terms (e.g., selection of all documents containing any three of five given index terms). With both files, weight factors can be used and calculated document weight factors can be part of the output. Also, the output can include the number of terms upon which selection was made, or a list of the terms, or both.

Both files can be used if the basic index terms of the request are to be expanded based upon term relationships included in the thesaurus, with or without weight factors assigned to the additional terms so generated.

Similarly, both can be used with expansion of the list of terms based upon "significant" associations of terms occurring in the file as a whole. Pairs, triplets, or larger numbers of terms may be used in the determination of association factors.

In the above two cases, both files permit limiting selection of documents to those meeting specified conditions of given and added index terms.

List-organized, but not the inverted, file permits additional search cycles using new index terms included in documents selected during the previous cycle. Here it is understood that the new terms are found solely because of their inclusion in documents selected on the basis of already-known terms. They are not derivable from the thesaurus. The new terms can be weighted and combined in these subsequent search cycles in the same manner as the original terms.

These search criteria involve data other than what are generally understood to be "index terms," but which may be incorporated into the search file.

<u>Dates</u> (year of publication, for example) can be a search criterion with both files. In the list-organized file, date is included in each document record. If so, it can be part of the serach output whether or not it is used as a selection criterion. In the inverted file, each time interval is set up as an index term record containing the numbers of all documents applicable. (This record almost always has thousands of detail entries.) Dates of selected documents cannot be provided, at least practicably, unless specified as a search parameter.

<u>Author's name, with an inverted file</u>, can be used only if it is an index term in the basic request and, further, only if the basic file has a record, for each author, with the list of pertinent document numbers. For practical purpose, it is not possible to determine the author of a document selected on the basis of other terms, even though the file includes the above record for each author.

Author's name, with a list-organized file, can be included readily as part of the output of all searches, provided only that it is a data field in each document field. In addition, the documents for each author can be "chained" into a list accessible through an enlarged entry table. If this is done, the search output can be expanded to include all other documents by the authors of those selected during the basic search.

Journal or publication name (usually coded), with a list-organized file, can be included as part of the search output in the same manner as author's name. Although this field also can be placed into lists, there is considered to be practically no advantage in doing so. With an inverted file, this field is subject to the same restrictions as author's name and, in practice, cannot be used.

Role indicators for index terms can be used with both files. Separate records (inverted file) or lists can be set up for each role-term combination; or, alternatively, a single record or list can be established for ther term, modifiers associated with the document number specifying the applicable role.

Link indicators definitely can be used with a list-organized file. Their use introduces several complexities with an inverted file, and it is not known if they can be incorporated efficiently. There is a good deal of controversy on the usefulness of link indicators in a document retrieval application. Analyses of their effects on file organization are not considered warranted at this time.

c. File Maintenance Differences. Updating a list-organized file requires more computing than an inverted file. The additional operations are those necessary to create the chain address for every index term in each new document. Inverted file updating is straightforward and simple: Create word-pairs for each new index term and document number combination, sort into (term) sequence and merge the document numbers into the existing term records. With serially assigned accession numbers, the merging occurs only at the end of each record to be updated; ideally, new numbers are added only at the end of the record. In general, the complete record for each index term in the new documents is read and completely rewritten. The operations are organized most efficiently in a sequential manner and even the sorting requires relatively little computer memory.

The most efficient algorithm for updating a list-organized file requires that the entire index term entry table be in the processor memory. If this is done, the chain addresses for each document can be created one after the other, the entry table being updated simultaneously, and the document transferred to the file storage medium before processing the next one. This approach uses a quite large amount of memory--two words or about ten characters--for each index term in the thesaurus and in many cases may not be practicable. Alternative methods take more computing time.

In the typical case of an updating cycle with about 500 new document entries, fewer file references are needed with a list-organized file. Although the entire entry table is read and rewritten, it is small compared to the document file itself. With a mass storage device, one access is required

for each document record processed; two may be needed. With magnetic tape storage, the file always can be organized so that new documents are added at each end (i.e., the file need not be in document number sequence) without rewriting the previously existing file. With an inverted file, a record access is necessary for each index term included in the input. For typical updatings with small document volumes, there usually are several times as many terms as documents. An inverted file requires more accesses to update the index file than does a list-organized file.

Periodic file purging (elimination of documents) is somewhat faster with a list-organized file than with an inverted file, provided that the purging involves a solid block of the oldest documents in the file. This is done so seldom--once or twice a year--that it is not an important factor in the selection of a file design. However, random purging also is not only possible, but simple, with a list-organized file. The storage space occupied by the record cannot be eliminated because of the need for retaining the chain addresses, but the document effectively can be "killed" by flagging or wiping out its number. Random purging can be done, but is not practicable, with an inverted file

d. File Storage Comparison. The exact method of setting up file records on the storage medium depends heavily upon the specifications of the storage device itself and the nature of data transfers to and from the central processor. It almost never is possible to optimize all of the several factors involved. Among the more important are: (1) Utilization of the data storage space available, particularly with mass storage devices; (2) effective, rather than instantaneous, transfer rates to and from the computer memory, especially with sequential-access storage; (3) access time, either sequential or random; (4) the amount of memory required for input/output data transfers in relation to the total available; and (5) the effects of file design on processing time. In practice, the detail file design is a compromise, each of several conflicting objectives being achieved in varying degrees (and, usually, none being fully realized).

In this respect, it should be noted that the degree of compromise necessary varies considerably for different types of approaches to basic file organization. With current mass storage devices, for example, it is considered much more difficult, if not impossible, to set up a list-organized file which will come as close to realizing its potential advantages as will an inverted file on the same device. A basic file organization which in theory may be superior or preferable to another may in practice be inferior or less efficie .

Because of the varying characteristics of storage devices and their interfaces with the rest of the processing system, it is appropriate to make only general remarks and comparisons on implications of the medium selected on the list-organized and inverted files.

If the index file is stored on magnetic tape or similar sequential-access devices, comparable efficiencies can be achieved with either type of organization. Tape blocks almost invariably are fairly long to attain high effective transfer rates and, with modern equipment, range from 500 characters up; larger blocks are desirable if enough memory can be allocated for input-output areas. Thus with both files, a number of records are packed into one

block. With an inverted file, the long records for common terms may be split into several blocks. The condition probably never arises with a list-organized file; 50 index terms for a document (the largest number report) creates a record on the order of 500 characters.

Two points may be noted. First, the list-organized file is about twice as large as the inverted and takes twice as long to process. Thus, if search criteria are within the scope of what it can handle, the inverted file is preferable when sequential access storage is used. Second, if a list-organized file is used, chain addresses must carry only in the forward direction of the tape. In practice, this results in mixing records of various sizes within the file. Unless the equipment includes a flexible input-output control word system (e.g., "scatter read"), time to search out individual records increases. Records in an inverted file can be grouped quite easily according to length (number of index terms included).

Three important characteristics affect the organization of a file on a mass storage device, such as a disc or drum. First, the random access capability requires specifying a record's location as a machine-fixed address--disc surface, track, and sector within track, for example. This factor causes no logical difficulty with either list-organized or inverted files; the machine addresses need not be the same as document numbers or index term codes. In a list-organized file, however, their use as chain addresses almost certainly increases the size of each record. This follows because the document number, which is what really is being chained, seldom exceeds six decimal digits, or 20-24 bits, while machine addresses of mass storage sectors usually take more bits then this.

Second, in many equipments, sectors have a fixed character capacity, usually in the 60-200 range, but sometimes larger. Data transfers may occur in one or more of three basic ways: (1) One complete sector at a time; (2) one sector, with the transfer terminated when the actual end of data is reached; and (3) multiple sectors, variable in number, transferred at a time. With both types of files, compromises are necessary to fit the variable-length records into fixed-length sectors and to handle long records which cannot be contained within a single sector. A few equipments provide for truly variable-length sectors, one sector terminating and the next beginning immediately after the end of each record. Thus one track can have a variable number of sectors, each of different length, track capacity setting the maximum sector size. This facility is well-adapted for files in which records are variable in length but, once established, are essentially static--i.e., do not expand or contract during subsequent processing. This is a basic characteristic of a document description and thus a list-organized file can readily utilize variable-sector storage.

Third, mass storage devices are relatively more expensive, per bit, than magnetic tapes and in operation the entire file must be available to the central processor. Thus it is desirable to utilize a high percentage of the available bit capacity for data storage. This may be difficult to achieve with fixed-length sectors and a list-organized file, where the maximum record may be 4-10 times as long as the minimum. Here it is doubtful if utilization of as much as 70% can be realized without sacrificing some of the potential advantages of this method of file organization. With variable-length sectors and one record per sector, the utilization may be somewhat better. Some

track capacity is used to record the machine sector addresses and other handware signals associated with variable-length data blocks. Normally, this is equivalent to many bits and, for the short records typical of document descriptions, may take 15% or more of the capacity potentially usable. In addition, the machine addresses tend to fairly long; if used as the chain addresses within each record, their greater length (than document numbers) further reduces the effective data storage capacity.

These factors are not so important with an inverted file, whose records increase in size with time and whose growth factor is taken into account in file design and storage allocation. Internal index term codes easily can be made the same as machine sector addresses and term records of like sizes can be grouped readily to utilize most of the capacity of fixed-length sectors. If variable-length sectors are used, the machine addresses take a much smaller percentage of track capacity because the average index term record is much longer than the average document description (a 7:1 ratio in the DDC sample and this probably is lower than in the typical document retrieval application).

e. Comparison of Search-Request Processing Requirements. Four factors affecting the processing of search requests may be noted: (1) Number of records accessed or acted upon; (2) amount of data transferred into the processor memory; (3) amount of computing necessary to determine the documents meeting the search criteria; and (4) the amount of memory required to hold data and the program.

It has been noted that, with an inverted file, one record is accessed for each index term in the request, some of them being very long. Their number seldom exceeds 20. With a list-organized file, the number of accesses is highly variable, but the individual records are short. The ideal search here is one in which the request contains an infrequently used term connected by a logical "and" relationship to all its other terms. Then only the documents in this one short list need be accessed. The case is not considered typical. The common term may not be infrequently used. The request may not be simple, but contain two or more subsets, each with one term having the desired "and" relationships. Or the selection criterion may be based upon partial matching against terms in the request. The "average" search against a list-organized file, then, requires traversing several lists and, although shortest lists can be selected whenever possible, the total number of records accessed is fairly large and several times as many as with the inverted file. It may also be noted that a variable percentage of records will be accessed and processed two or more times because they belong to more than one of the lists involved. Consequently, total access time--in the 15-75 millisecond range for typical mass storage devices--normally is several times as long with a list-organized as with an inverted file. This is an important design consideration for a real-time document retrieval application.

The amount of data transferred into the processor is the product of the number of records accessed and their average length. In a list-organized file, the average length of records examined is about the same as that of the file as a whole. This is not true of an inverted file. An examination of a number of requests and some published data on this aspect indicate that the average length (number of documents) of search terms is considerably larger than that of the index terms in the file as a whole. This is

tantamount to saying that search requests typically contain several rather common terms. (In a list-organized file, this means that the average length of the lists in a request are greater than that of the total file.) No definitive data have been obtained as to which type of file organization results in the transfer of the lesser amount of data. However, an answer to this question may not be of major importance. With most current equipments, data transfers occur at very high speeds. With mass storage devices, access time for a record greatly exceeds the actual transfer time of all except extremely large blocks of data.

Except for control and input-output programming, the computing time necessary to process a search request is largely a function of the number of comparisons made. This is easily determinable with an inverted file in which the comparisons are made against the sequenced list of document numbers in the record for each index term and a similarly ordered list of document numbers meeting the search criteria to the current point of processing. The number of comparisons effectively is the same as the total of the document numbers read in with all index terms in the request and is independent of the order in which the terms are processed. (Actually, it is a little less, because the two lists usually are not exhausted simultaneously.)

With a list-organized file, the number of comparisons is not easily predictable. All pertinent index terms in the request must be examined and pass the search criteria to accept a document. It is rejected at the first failure to pass a selection criterion and this occurs after examining a variable number of index terms. No reports of analyses into this phase have been seen. Second, and more important, the number of comparisons is highly dependent upon the <u>order</u> in which the index terms are processed. Within each document record, the terms are in some prescribed order, which without loss of generality can be assumed to be ascending sequence on index term code. Unless the terms in the request can be taken in the same sequence, the record may be scanned several times to find individual terms, each scanning involving several comparisons. It is considered probable that the request terms can be so ordered, but the comparisons subroutines probably are longer, and take more computing time, than the straightforward "accept-reject" possible with an inverted file.

The program for processing search requests against an inverted file appears to be less complex than that for a list-organized file and thus to require a somewhat smaller amount of computer memory. However, the inverted file needs much more memory for data stroage. If list organized, each document record is accepted or rejected on the spot and no intermediate data are carried over from one to the next. If inverted, an intermediate list of document numbers is carried over to each successive index term and memory must be allocated to hold it. This list may be fairly long--several hundred documents at some stages of the processing--or there may be more than one of them, depending upon the logical complexity of the request and the order in which the terms are processed [illegible] addition, with a mass storage device, its data input area is large, be[illegible] it is necessary (or at least highly desirable) to provide for reading successive blocks of several hundred words each for index terms appearing in many documents. On the other hand, the input area with a list-organized file only need be large enough to handle the longest document description. If magnetic tape is used, the blocks are about the same size with either file and the input areas therefore are comparable.

With batch processing of search requests against an inverted file on magnetic tape, intermediate data storage requirements often are so large that the processing of the main file is limited to writing out a "work tape" of the records for the index terms involved. Subsequently each request is processed, one after the other, against this small "work tape." Batched searching against document-sequenced or list-organized files can be done as each successive record is read in, although the latter type of organization may introduce a rather complex control program to handle the multiple lists being followed. Use of mass storage devices eliminates this type of batch processing; each request is acted upon individually even if several are received at one time.

3. Determination of Optimum File Organization for Document Retrieval.

From the discussion of the previous two sections, it is considered that the inverted file is the more efficient organization if the types of searches it can accept and the output data it provides meet the application requirements. This is true for both sequential and random access file storage. The file is smaller than any other except the straight document-sequenced organization; is simple to maintain; requires fewer record accesses in processing a search request; probably selects documents with considerably less internal computing; and is susceptible to efficient operation with either sequential or random access types of file storage.

The basic disadvantages of the inverted file relate to the scope or complexity of search criteria which are permissible and to its restricted output in response to search requests. Although it may be granted that the inverted organization adequately meets the requirements of many, if not most, existing document retrieval applications, there appears to be a definite trend toward more complex and sophisticated search criteria and more data, short of abstracts, in the output. These are inevitable--and, on the whole, desirable--tendencies for an application which has a relatively short history of mechanized processing. Progressively increasing complexity and sophistication have typified virtually every application converted to electronic processing systems, and there is no reason to think that document retrieval is any different. As a matter of fact, it is doubtful if there is much justification for such a system if it accomplishes no more than can be done, for example, with "peek-a-boo" cards.

Many of these ramifications are based upon data either already contained, or easily included, in files with document-oriented records. Also, they often are directed toward an ultimate real-time operation requiring random access to file records and, at some point, remote query-display devices and the resultant ability of the requester to control and modify the handling of his query as a part of its processing.

The question then arises: Is the list-organized file the most efficient method of storing a document description file when the inverted sequence will not meet the requirements of the application? After careful analysis and evaluation of the factors and implications involved, it is our opinion that the answer must be an unqualified "No." If a list-organized file meets the processing requirements of a document retrieval application, then a conventional inverted file together with a conventional document-sequenced file constitutes a more efficient and preferable form of data storage.

This statement is not particularly difficult to substantiate. In fact, the suggested organization is a direct and immediate product of analyzing a list-organized file and its processing implications. Much of the rather voluminous literature on this method of file organization seems to assume that it is a new methodology and is devoted to the design, use and manipulation of lists. This approach has been made possible by adding large-capacity, random-access storage devices to the electronic data processor, the complete system removing the necessity for essentially sequential processing which characterizes earlier types of data processors. Too little attention has been paid to what a list-organized file really is or to the conditions under which it may be the optimum form for storing data to be processed.

File organization and design always have been predicated upon the media available for data storage, the processing to be done upon the data and the characteristics of the "tools" available to do the processing. They still are. These three factors are heavily interdependent. The principle of the list-organized file is not new, but its manifestations and method of use differ, of course, when random rather than basically sequential access to records becomes possible.

The closest counterparts to list-organized files are found in those processed manually, where at least quasi-random access is possible. (Technically, access to discs and drums also is quasi-random.) One of the oldest is the list of synonyms and antonyms given for many words in a dictionary or thesaurus. This is a direct counterpart; the cross-references are chain addresses leading to other file records having something in common with the current one. Somewhat less obvious is the widespread use of colored flags or inserts in visible record or vertical files to identify records possessing a similar attribute value; moreover, one record can belong to several different lists. In this case, the flag merely identifies a record having a specific attribute and does not "chain" to the next record in the list. There is a difference in technique arising because of the particular characteristics of the file storage media and the manual processing against it. It makes possible the processing of all records on a "list" on a quasi-random basis and without the necessity of examining every entry in the file. This is exactly the objective of a list-organized file in an electronic computer application. The use of edge-notched cards makes possible an approach logically the same as that described above and adds a degree of "mechanization" to finding the records in one list.

There is no close counterpart to list organization in processing systems based upon punch cards or embossed plates as the file storage medium. This arises because the various equipments found in these systems handle files purely on a sequential basis. Maintaining two cards of the same basic data in different sequences is somewhat analogous, the filing keys of one desk corresponding to lists into the other.

Records in a list-organized file can be accessed in one of two ways. First, they can be located by the keys upon which the file is sequenced, each record being in a specific location relative to all others. A record may be found either by sequential search of the file or, if the storage and processing system permits, on a random access basis. Second, records having some attribute in common can be located by entering the list for that attribute and, using the chain addresses or tags, finding each related record

in sequence. In practice, the technique is confined to systems permitting essentially random access to any desired record.

A record in a list-organized file contains two types of data fields. First are those which pertain to the record itself--in a document file, these are the index terms, author, journal, date of publication, etc., which describe a given document. Second are the chain addresses, each of which links the record to another one having the same attribute value for the data field linked. These chain addresses do not pertain to the record and add nothing to the information contained in the first type of data field. Elimination of all chain addresses in the file removes absolutely no information; all it removes is one method of entering it.

Assume there exists a list-organized index file for document retrieval, with document numbers as chain addresses. The entry word for index term A (List A) gives a document number containing A. This document record in turn includes a chain address which is the number of another document containing A; and so on, the chain address of the last document in List A containing a unique code signifying "end-of-list." All of the chain addresses linked from the entry word for index term A can be removed from the file and set up as a record for A. What is the nature of this record? Index term A followed by all document numbers in which it appears. This is exactly the record for index term A in an inverted file.

The process of removing chain addresses from the list-organized file and creating index term records can be repeated for all terms in the entry word table. Upon completion, the file has been split into two parts. The index terms and the chain addresses constitute a normal inverted file. The original list-organized file, now with all chain addresses eliminated, is a normal document-sequenced file. Thus, <u>a list-organized document retrieval file is a direct merger of the conventional document-sequenced and inverted files</u>. Specifically, it is a document-sequenced file to which has been added, as chain addresses, the index term records of the inverted file.

The combination of an inverted and document-sequenced file is one alternate way of setting up exactly the same information as is contained in a list-organized file. Because an inverted file record not only corresponds to, but also is, a list of chain addresses, it can be used exactly as they are used in a list-organized file. There is no mandatory reason for a record in the file to contain the chain address of another one in the list. The list of chain addresse. can just as well be successive entries in a separate record. The inverted and document-sequenced files permit carrying out any type of processing possible with alist organization and, in addition, enable execution of operations peculiar to the inverted sequence.

This dual file has several advantages over the list-organized one:

File updating is simpler and faster. It is unnecessary to perform the operations required to insert chain addresses within a single file.

Search comparisons can be based upon index term operations in the normal manner of the inverted file. This requires access to only a few records and, usually, less computing than operating on lists. The complete records for selected documents must be obtained from the other file, but the total number of accesses almost always is much less than with the list-organized file.

Searches can be conducted against documents in lists if considered appropriate or faster. By incorporating suitable criteria, such as presence in the request of an infrequently used index term, the search program can be modified to select the type of search which probably will be completed fastest or most efficiently.

Searches against index term lists transfer less than half as much data into memory as the conventional list-organized file, because there are no extraneous chain addresses in the document-sequenced index file. The chaining itself also is simpler and faster; the next document number is in a known location in the inverted file record which serves as entry, rather than in an unknown position in the record currently being processed.

If desired, searches can be a combination of the inverted and list-organized approaches. That is, comparison of index term records can continue until the number of documents so far meeting the criteria is small, at which time document records can be scanned. The intermediate group of document numbers serves as the entry list.

The possibility exists of organizing the document-sequence file in a manner which will reduce the access time to its records. This arises because all document numbers in a list, or selected in processing the search request, are known before any of them are accessed. If the records are suitably organized on the mass storage device, the order of picking up records can be chosen to reduce the average access time well under that possible with a random search.

The advantages and flexibility of the dual file technique indicate that it is a preferable and more efficient approach than the conventional list-organized file. Detailed analysis of the use of the dual file to process lists has revealed only one disadvantage, considered to be of minor importance: More memory must be allocated to hold the document numbers or other identifying keys of the records in the list. In practice, long lists of keys would be subdivided and several accesses made for the complete list. At 50 keys per subdivision, the dual file approach requires 2% more record accesses than does the list-organized file.

Although this study of the implications of the list-organized file has been conducted with specific reference to a document retrieval application, the conclusions apply to many other applications in which it is a possible method of file organization. The document index file differs from most other business data files in two significant respects. First, a document description record once established in the file remains static and unchanged until finally it is removed completely. Its field entries do not change and its length does not vary by the addition and deletion of temporary "trailer" data.

Consequently, the lists to which it belongs remain fixed. Also, the lists themselves change only as documents are added to or deleted from the file, not from processing actions on records already in the file. Changes in field entries and variations in "trailer" data are normal occurrences in processing most other files and the lists to which a record belongs change, or can change, as a result of routine processing. Second, most of the references to a document description file are not made on its identifying and sequencing key (document number), but upon an attribute value (index term) it contains. Again this is atypical; most files have many references based upon indexing keys and relatively fewer upon attribute values.

A parts fil used for stock and inventory control purposes is a typical example of a business-type data file. Some military activities, at least, have established parts files in list-organized form and are processing against them. Because many of the processing actions are routine orders for or receipts of material, the file is established in part number (or stock number) sequence and, in these common cases, access to a record is through this filing key. However, a veriety of other demands are placed upon the file. Typical examples are: All parts used in a given equipment; all parts obtainable from a specified supplier; all parts currently on order; all parts with a cost of $1.50-$1.99; and all parts whose stock position is below their established low limits. Records with attributes of these types obviously can be chained together in a list-organized file. In many cases, the required output of processing a list is more than the part numbers and access to all or a portion of their file records is necessary.

It is considered that a list-organized parts file is less efficient and not preferable to a dual file. The latter is easier to maintain and update. The routine processing actions transfer shorter records because there are no superfluous chain addresses in the part number file itself. Many of the lists are referenced at relatively infrequent intervals and the chain address records might be stored more economically on a medium less expensive than a mass storage device. It is conceded readily that the more efficient processing and lesser computing time attainable with the dual file may be more potential than realizable. Access time to records may dwarf actual data transfer and computing time and this may make any time saving relatively insignificant. There is no practical advantage of devising a more efficient system unless productive use can be made of the time or memory saved, or unless comparable results can be achieved with a smaller amount of hardware.

Nonetheless, it does not appear unreasonable to expect that the list-organized file compete and be evaluated on its own merits against alternative methods of data storage and processing. Tacit assumption of its efficiency without recognizing its disadvantages can lead to using list organization in applications where other approaches may result in markedly lower time or cost of processing. The list-organized file unquestionably has a role in modern processing systems. It is highly desirable to analyze and delineate the conditions under which it--and other forms of data organization--can be used most efficiently.

4. Detail Design of Inverted and Document-Sequenced Files.

This section proposes a basic method of approach for the most efficient detail index file design in a document retrieval application. It takes advantage of data characteristics which can be used to minimize any one or more of record access, data transfer or internal computing time. Although the discussion assumes that files are maintained on mass storage devices, the inverted file design also can be used advantageously with magnetic tapes.

Any detailed file design depends heavily upon the specifications of the storage unit and its computer interface. Because these vary widely, only the general approach is outlined. Modifications are necessary to fit the general method into the framework of a specific equipment configuration.

a. Design of the Inverted File. This file typically is set up in sequence on index term code and in document number sequence within the record for each term. Many search requests contain several fairly common index terms with several hundreds or thousands of document numbers each. Even in libraries of modest size and average depth of indexing, a typical search may involve on the order of 10,000 of these, each of which must be transferred into memory and enters into a comparison loop. Quite commonly, a small group of, say, 20 documents, selected on the basis of comparisons so far made, is matched against an index term with 2,000 entries--frequently followed by other high-usage terms.

If the index term record with 2,000 entries could be broken into 200 subsets, for example, of about 10 documents each, then the 20 intermediate document numbers could be processed by accessing not over 20 of these subsets and making about 200 comparisons, eliminating 90% of the word transfers and comparisons otherwise needed.

Four basic system requirements should be met if an inverted file is to be organized successfully in this manner:

(1) The document number itself must determine the subset to which it belongs.

(2) Each subset should contain close to the same average number of documents.

(3) The data should utilize a reasonably high percentage of the potential capacity of the storage device.

(4) The system should provide for increasing the number of subsets as more documents are added to the index term record. It should be self-organizing in the sense that the computer program includes criteria permitting automatic adjustment of the number of subsets as documents are added to or deleted from an index term.

In addition, a fifth requirement exists if the storage device cannot handle variable-length records; it is closely related to (3):

(5) With variation in the number of entries, overflowing the capacity of a subset is possible. The technique should permit determining the subset size necessary to give statistical assurance that the probability of overflow does not exceed some arbitrary low value.

These requirements indicate at once that some randomizing technique on a document number is a possible means of determining its subset and, for all documents in an index term record, giving a statistically-predictable distribution of the number of entries in each subset. A simple randomizing scheme is suggested. If document accession numbers are assigned in ascending numerical sequence--this is the most common method--then the well-known method of "terminal digit" filing effectively provides the desired randomizing. For practical purposes, each of the number 0-9 is the terminal digit of exactly 10% of the documents in a library. There is no reason to assume that the usage of an index term is in any way related to or dependent upon the terminal digits; i.e., there is a probability $p = 0.1$ that any given document using the term has an accession number terminating in 3, or any other decimal digit. If the term is used in N documents, the average number in each of the ten subsets 0-9 is, of course, pN and the standard deviation is $\sigma = \sqrt{pqN}$.

Terminal digit studies have been made of a number of index terms in the DDC sample and several analyses conducted on two 10% subsamples consisting of document numbers ending in "2" or "9." None of these give any statistical reason to doubt the randomness of index terms and the terminal digits of documents. Creating subsets based upon terminal digits, then, is a statistically valid approach which will distribute entries into them in approximately equal number and with a predictable standard deviation from the average.

Terminal digit filing is not new in document retrieval. It has been used for many years in manual systems, particularly those based upon the well-known "Uniterm" concept. Here document number commonly are entered in ten columns, based upon the terminal digit.

Use of decimal terminal digits to determine subsets has some practical disadvantages. If the number of documents posted to an index term increases to the point where more subsets are desirable, then adding the next higher terminal digit (the "tens" to the "units," for example) multiplies their number by ten. Also, each new subset has only one-tenth as many entries, on the average. Fewer subsets could be created by using ranges of numbers; e.g., increasing 10 subsets to 20 is possible by grouping on terminal digits 00-04, 05-09, etc. However, entry to the proper subset is somewhat more complicated.

A preferable approach is to convert the decimal document number to binary. Each bit added as a terminal digit doubles the number of sets and halves their average number of entries. Many, but not all, electronic processors, have binary arithmetic capabilities and, possible of even more importance, sector addresses of many mass storage devices are in binary form.

Suppose an index term record contains 16 subsets, determined by and sequenced in order on the four binary terminal digits 0000 through 1111. The location of the entire record on the mass storage device is determined

through the index term code. Desired subsets are specified by the terminal bits of a document number and are in a known position relative to the first subset 0000. Consequently any specified subset can be accessed readily, provided the number of subsets is known. This may range from a single subset for infrequently used index terms up to several thousand for the highly common ones.

The most efficient technique so far found interprets the storage unit address or addresses for a record in the general form nAs, where

n is a 4-bit prefix specifying the number of subsets 2^n (i.e., 1, 2, 4, ..., 32,678);

A is the storage unit sector address of the first subset or group of subsets for an index term; and

s is an increment to A such that A + s either (1) is the storage unit address for subset s if there is one subset per sector, or (2) specifies the sector and subset within sector if subsets are grouped 2^i per sector.

nAφ is stored as the entry table address for each index term in the inverted file. Preferably, it is part of the mechanized thesaurus, where it is readily available at the time the terms of the search request are validated.

Data transfer and comparison times are small when there are only a few entries in the average subset. Minimizing these times conflicts with the objective of utilizing a reasonable percentage of potential mass storage capacity. For example, if an index term record with $N = 2^n$ is broken into $\frac{2^n}{4}$ subsets with an average of four entries each, then

$$\sigma = \sqrt{\frac{1}{2^{n-2}} \cdot \frac{2^{n-2}-1}{2^{n-2}} \cdot 2^n} = \sqrt{\frac{2^{n-2}-1}{2^{n-4}}} \approx 2.$$

If the subset size is fixed at 8 words, the storage utilization is only 50% and there is a $p \approx 0.025$ that a subset will overflow; that is, on the average about one out of 40 subsets can be expected to have more than 8 entries. Somewhat better storage utilization might be realized with variable-length sectors, but the fixed hardware requirements still are a fairly large percentage--possibly 30-40%--of potential capacity.

Larger sectors result in better storage utilization but also increase data transfer and computing times. If $N = 2^n$ and $\frac{2^n}{16}$ subsets are set up, with an average of 16 entries each, then

$$\sigma = \sqrt{\frac{1}{2^{n-4}} \cdot \frac{2^{n-4}-1}{2^{n-4}} \cdot 2^n} = \sqrt{\frac{2^{n-4}-1}{2^{n-8}}} \approx 4.$$

Here a fixed subset size of 24 words yields 67% storage utilization with the same p = 0.025 overflow probability. If variable-length sectors are permissible, utilization of 90% or more should be possible.

The conflicting objectives of small subset size and reasonably high utilization of storage capacity are resolved on the basis of characteristics of the equipment to be used and administrative determination of acceptable utilization.

In the subdivided file, index terms are grouped by number of subsets included and ordered in ascending sequence on this number. The first group consists of terms appearing in a single document, a sector of n words containing n terms. Index terms with 2, 3, 4, ... usages similarly are grouped and packed several per sector; the sequence of document numbers within each term is on terminal bits. This grouping continues until the number of usages is enough to warrant creation of two subsets and splitting documents into two groups based upon the terminal bit. With sectors of eight words, analysis of the DDC sample indicates that the split can begin with terms having 5 to 6 usages. The first groups of terms then have this format.

1 Usage: 8 index terms per sector

2 Usages: 4 index terms per sector

3 Usages: 2 index terms per sector

4 Usages: 2 index terms per sector

For these, the machine address carried in the entry table in the form aNs is interpreted thus:

a: Number of usages of index term;

N: Mass storage unit address of sector containing the term,

s: Relative number of term record within sector.

The remainder of the index terms are established initially in the minimum possible number of sectors. Thus, still using 8-word sectors, all terms with 5-8 usages always can be stored in two sectors, based upon "0" or "1" as terminal bits. Most terms with 9-12 usages also can be, as can some with 13-16, the probability of overflow increasing with the number of terms. If overflow occurs, the number of sectors is doubled and the assignment of document numbers made on the basis of two terminal bits--00, 01, 10 and 11. Thus, although the 2σ level is used to determine sector capacity and the probability of overflow, the latter is not allowed to occur.

Most terms with up to 22-24 usages can be contained in four sectors, as can some with 25-32. Whenever an overflow occurs, the number of sections again is doubled and another terminal bit added for sector identification. This cycle is repeated until all index terms have been set up in the subdivided inverted file. Each term is placed in the minimum number of sectors for which no overflow occurs.

As new documents are added to the file, they are entered in the proper sector for each index term. Whenever a sector for a term overflow, their number is doubled and the record is transferred into the next higher group on the mass storage device. Simultaneously, the machine address in the index term entry word is changed to the new location. Thus the updating program continuously reorganizes the file as sector subdivision becomes necessary, the movement always being toward a larger number of sectors.

The basic procedure can be applied for any desired sector size and percentage utilization of the mass storage unit capacity. The systematic breakdown of document numbers permits searches to be localized within specific sectors determined by the numbers of the documents which have met the criteria up to the current stage of processing.

It may be noted also that this technique of terminal digit filing can reduce significantly the theoretical number of bits required to hold the inverted file. When a sector contains only documents which have the same s terminal bits, then they become redundant and need not be retained in the stored record. For frequently used index terms, where $s \geq 7$ or 8, these potential savings exceed 30% of the number of bits in a document number and, for very common terms, may approximate 75%. Thus either more documents can be stored in a sector of given bit capacity or, alternatively, a constant number of documents stored in fewer bits. With existing equipments and mass storage units, this potential saving probably cannot be realized.

b. Order of the Document-Sequenced File. If document-sequenced file is used in conjunction with an inverted file, access time to document records can be minimized if they are grouped on terminal digits. Suppose, for example, that the tracks on a disc or drum are broken into 16 major sectors, numbered (in binary) from 0000 to 1111. Each document record is stored in the major sector determined by the four terminal bits in the document number.

Because documents in the inverted file are sequenced and processed in this same order, any list of document records to be accessed also is in this order. Therefore up to 16 separate records can be transferred to the processor memory during a single revolution of the drum or disc storage unit. A random search for the same documents would be at average rate of only two per revolution. Ordering of the document records on terminal digits thus eliminates a large percentage of this average access time.

* * * * * * * * *

It is concluded that the combination of an inverted and document-sequenced file is a more efficient type of organization than the conventional list-organized file. In addition, this dual file can be set up to reduce both the processing time in handling a search request and the time required to access complete document records. These advantages cannot be realized with the list-organized file.

II. INDEX TERM ASSOCIATIONS IN THE DDC SAMPLE

Creation of the data files to simulate the operation of the Multi-List System resulted in the formation of all pair associations among the 599 most common DDC descriptors. In addition, some other association statistics have been developed during the statistical analysis of the characteristics of this sample file. Some of the results are presented in this section. The discussion is not a comprehensive study of pair associations and their uses in a document retrieval application.

A. ASSOCIATIONS AMONG THE 599 MOST COMMON DESCRIPTORS

1. Occurrences of Pair Associations.

The 599 descriptors have 49,306 different pair combinations--27.6% of the number possible--with 248,425 occurrences, an average of almost exactly five each. 41% of the pairs occur only once and almost 80% five times or less. Only 2% of the pairs appear 33 times or more, but they represent 25% of total occurrences. Table A-2 (Appendix A) summarizes the distribution of pairs by number of occurrences. The cumulative percentages of different pairs and total occurrences also are shown graphically in Chart 5.

The entire 38,402-document sample has about 209,000 different pairs with 530,800 total occurrences. The 10.7% of descriptors comprising the 599 most frequently used generate 24% of the different pairs and 47% of the occurrences. The remaining 89.3% of descriptors in the sample create about 160,000 different pairs with 282,400 total occurrences, an average of only 1.77 occurrences per pair. Evidently, in the sample as a whole, multiple occurrences of pairs are in the minority.

2. Different Pairs and Occurrences Among the 599 Descriptors.

It has been noted that the number of different pairs decreases with frequency of usage among the 599 most common DDC descriptors. This question naturally arises: Is there any close correlation between the number of different pairs created and the total occurrences of those pairs? Table A-3 (Appendix A) shows the distribution of the 599 descriptors against these two factors as coordinates. Although it indicates a general correlation, the distribution is marked by wide variations. In general, descriptors creating relatively few different pairs have fewer average occurrences per pair than those with many. However, for any one range of numbers of different pairs, average occurrences for different descriptors usually vary by factors of three or four to one.

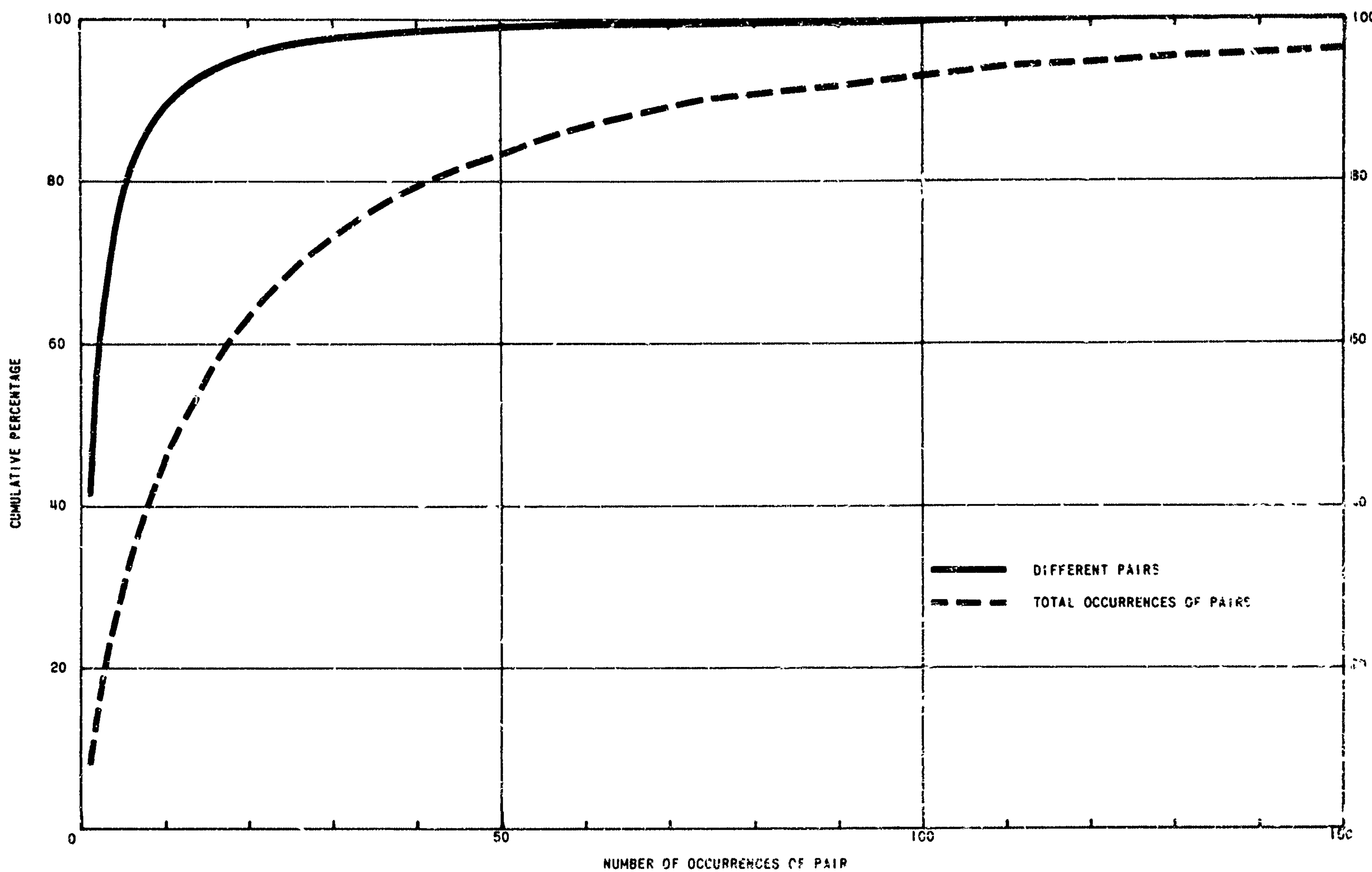


Chart 5

599 Most Common DDC Descriptors: Cumulative Percentages
of Pair Associations Classified by Total Occurrences of Pairs

3. Association Factors for Pair Occurrences.

One measure of association is $p(B|A)$, the probability of occurrence of descriptor B in a document, given that it contains descriptor A. The two permutations of a pair result in two such probabilities, which in general are different: $p(A|B) \neq p(B|A)$.
Let

f = Number of occurrences of the pair, descriptor A with descriptor B;

F_A = Frequency of usage of descriptor A alone;

F_B = Frequency of usage of descriptor B alone.

Then

$$p(A|B) = \frac{f}{F_B} \quad \text{and} \quad p(B|A) = \frac{f}{F_A} .$$

Table A-4 (Appendix A) summarizes the values of p for the 98,612 pair permutations among the 599 most common descriptors, almost two-thirds of them have $p < 0.015$. Only 61 have $p \geq 0.50$. Of these, only two are permutations of the same pair: "Peroxides" (A) and "Hydrogen Compounds" (B), for which $p(A|B) = 0.75$ and $p(B|A) = 0.64$. ($f = 56$; $F_A = 87$ and $F_B = 75$.) It may be noted that in a hierarchal descriptor relationship, "Peroxide" would be expected to fall into the class of "Hydrogen Compounds" and thus the probability of occurrence of the latter, given "Peroxide" as being present in a document, should be greater than the converse relationship. Actually, the reverse condition exists. No meaningful conclusion is apparent.

For 11 of the remaining 59 permutations with $p \geq 0.50$, the converse probability is between 0.20 and 0.50; the rest range downward to 18 for which $p \leq 0.02$. Further, for 40 of these 59, the second descriptor--the one whose probability of occurrence is given by p--is one of six very common ones. Design (Rank 1); Tests (2); Guided Missiles (5); Radar Equipment (17); Polymers (36); and Projectiles (37). For most of these, the converse probability is quite low. This is to be expected; these common descriptors appear in thousands of documents compared to a few hundreds at most for the other member of the pair. For example, "Cargo Vehicles" (Rank 526) appears in 82 documents, 51 of which also contain "Tests." Thus, $p(\text{Tests}|\text{Cargo Vehicles}) = 0.62$. "Tests," however, is used in 5,237 documents and therefore $p(\text{Cargo Vehicles}|\text{Tests}) = 0.01$.

4. Associations of 50 Most Common Descriptors.

Table A-5 (Appendix A) details the number of pairs and total occurrences for each of the 50 most common descriptors. Associations are broken down into those with the 599 most common and with the remaining 4,941 descriptors. This table again makes it apparent that several occurrences of a pair are the exception, even when one member is common. (The 50 most common descriptors are used in 443 or more documents; the 4,941 less common have 71 or fewer usages.)

B. DESCRIPTOR ASSOCIATIONS AMONG DDC GROUPS AND FIELDS OF INTEREST

The summaries described here are based upon the 292 groups and 19 fields of interest described in the ASTIA thesaurus, 1960 edition, applicable during the time period covered by the sample. There now are 33 fields.

1. Most Common Descriptors Summarized by Field.

Table A-6 (Appendix A) summarizes the 599 most common descriptors into ASTIA fields, together with the number of pair permutations having one or both members in the field and their total occurrences. Some fields and groups are richly represented; others have few descriptors among these 599. This variation reflects the types of documents in the sample and, by extension, the relative distribution of document acquisitions by fields of interest. Although the thesaurus must provide for adequate indexing of documents in all fields of interest, descriptor usage is a function of the types and numbers of documents received. Descriptors in fields represented by many documents not only have many chances to be used, but also many chances to create different pairs and multiple occurrences of one pair.

2. Associations Classified by Group and Field of Interest.

It is desirable to test the hypothesis that the DDC thesaurus has a hierarchal structure which is reflected in descriptor associations and which can be used as a tool in formulating search requests.

For this purpose, the pair associations formed by the descriptors in each of the 155 groups have been summarized and classified by all of the other groups to which the second descriptor of each pair has been assigned. Each group, A, is represented by a single summary page which lists every other group B_i, having descriptors associated with those in A. Three quantities are accumulated for each of B_i entries: (1) Number of different descriptors in group A entering into associations with those in group B_i; (2) number of different pairs formed; and (3) total occurrences of these pairs. In addition, the last two quantities are totalled for each of the 19 major fields of interest into which the 292 groups are combined. Table A-7 (Appendix A) shows a typical page of this summary; it is for group 145 (Materials) in field 10 (Materials and Metals).

55 of the groups, or 35%, have only one descriptor each and another 30 have two. 13, or about 8%, include ten or more descriptors. The number of other groups with which associations occur averages 93.5, about 60% of the number possible. The range is from 36 (Drugs and Biologicals, group 072, with one descriptor) to the maximum of 154 for General Concepts, group 292, with 15 descriptors. There is a definite correlation between the number of descriptors in a group and the number of other groups involved in associations. The 55 groups with only one descriptor each form associations with an average of 66 other groups; the 13 with ten or more descriptors average 141.5 each.

Table A-8 (Appendix A) summarizes, by fields, the frequencies of pair associations, together with the number of occurrences for which both descriptors are in the same group or the same field-of-interest. Co-usage of

two descriptors in one group represents only 1,788, or 1.8%, of the number of different pairs and 4.5% of total occurrences. Although seemingly low, this is over 75% of the possible number of intragroup pairs. For most groups with 2-4 descriptors, all possible pairs actually exist, the percentage occurring decreasing slowly (and not uniformly) as the number of descriptors in the group increases. Only four of the 100 groups with two or more descriptors have no intragroup pairs, all four have either two or three descriptors. Thus if two of these 599 most common descriptors are in the same group, there is a high probability that they will be associated in use. Furthermore, they are likely to occur 2½ times as often as other pairs. However, intragroup associations are only a relatively insignificant part of all of them.

Although they account for only 11% of the number of different pairs, 51% of the intrafield associations which can exist do occur in the sample--9,582 of a possible 18,558. Actually, 17 of the 19 fields exceed this percentage and 9 have more than 70% of the possible pairs. The over-all average is heavily weighted by the 133 descriptors in Physics and Mathematics; only 3,673 (42%) of the 8,778 possible do exist and 47% of the potential number is concentrated in this one field.

Interfield associations predominate among these 599 descriptors. Table A-9A (Appendix A) summarizes these interfield usages by numbers of different pairs and Table A-9B by numbers of occurrences. (Entries in the body of these tables are symmetrical about the underlined diagonal.) All possible combinations exist except for Bio-Sciences with Civil Engineering or Propulsion Systems. As might be expected, all fields form many associations with descriptors in Applied Research, Miscellaneous Arts & Sciences and Physics & Mathematics. Table A-9C shows the number of associations actually existing as a percentage of the number possible.

The foregoing comments can be summarized briefly. Among these common descriptors, there is a 0.25 probability that any two taken at random will be associated in use. If the two are in the same DDC field, the probability of co-occurrence is doubled; if in the same group, tripled. On the average, almost 90% of the different pairs and 85% of total occurrences involve descriptors in two fields. Pairs within the same group have a markedly higher average number of occurrences than other pairs; those within one field have a somewhat higher average. All of these data have been based upon an analysis of the 599 most common descriptors in a file of 38,402 documents, each descriptor occurring in 72 or more of them.

Whether or not these results indicate any tendency toward a "hierarchal structure" in descriptor associations is somewhat uncertain. Although intragroup and intrafield associations of descriptors are much more probable than the others, and occur more often, it seems questionable to base a hierarchy on 10% or less of different pairs and 15%, at most, of occurrences. Interfield associations of descriptors are predominant. Furthermore, frequently occurring pairs are the exception. 41% occur only once, 79% five times or less, and half of all occurrences are accounted for by pairs appearing 12 times or less.

3. Pair Associations Among All Descriptors.

Talbe A-10 (Appendix A) summarizes pair occurrences among all descriptors in the sample, classified by the number of usages of descriptors. The 5,540 descriptors in the 38,402 documents form 418,400 pair permutations with 1,061,600 occurrences, an average of only 2.5 each. It is estimated that over 80% of the pairs in the sample occur only once or twice each.

C. COMMENTS ON STATISTICAL ASSOCIATION MEASURES

Many of the association measures which have been proposed are based upon the conventional 2-way contingency table, or can be expressed in terms of its cell entries:

	I	II	Total
1	f	B - f	B
2	A - f	N - A - B + f	N - B
Total	A	N - A	N

where

A: Number of documents described by an index term D_A.

B: Number of documents described by an index term D_B.

f: Number of documents described by both index terms D_A and D_B.

N: Number of documents in the library.

Occasionally, it is desirable to consider the total occurrences of all index terms, both singly and in pairs. This notation is used:

A_i: Number of documents described by D_i.

$f_{i,j}$: Number of documents described by both D_i and D_j.

c: Number of different index terms, D_i, used in a document.

$\sum A_i = \sum_{1}^{c} A_i \left[= \sum B_j \right]$: Total number of occurrences of all index terms.

$\sum f_{i,j} = \sum_{i=1}^{c} \sum_{j=1}^{c} f_{i,j}$: Total number of occurrences of all pairs formed by all index terms $D_i D_j$.

In the DDC sample, the number of occurrences A_i of any random index term D_i usually is very small compared with N (= 38,402). Of the 5540 different descriptors represented, only 58 occur over 400 times; i.e., for 99% of the descriptors, $A_i < 0.01N$. For 80% of them, $A_i < 0.001N$. Because $f_{i,j}$ cannot exceed the lesser of A_i and B_j, it follows that, in most cases, the magnitudes of f, A - f and B - f in the contingency table are small compared with the fourth entry, N - A - B + f. Although comparable data for other applications have not been seen, it appears probable that most of them will be somewhat similar in nature to that of DDC, possibly with smaller percentages of index terms at the 0.01N and 0.001N levels--95-98% with $A_i < 0.01N$ and 40-75% with $A_i < 0.001N$.

1. Association Measures.

Among the first measures of association proposed were three by Maron and Kuhns [10], who developed them as part of a more general statistical approach to the problem of document retrieval. The first is the conditional probability that, if the term D_B is assigned to a document, then D_A also is:

$$P(D_A|D_B) = \frac{f}{B} . \qquad (1)$$

The second is the inverse conditional probability of (1); i.e., if D_A is known to be assigned to a document, then D_B also is:

$$P(D_B|D_A) = \frac{f}{A} . \qquad (2)$$

This actually is not a second relationship, but the first with D_A and D_B interchanged in meaning. However, its differentiation is desirable, because in general $P(D_A|D_B) \neq P(D_B|D_A)$ and, in fact, is equal only if A = B, which is not often the case.

$P(D_A|D_B)$ ranges in value from zero (f = 0) to 1 (f = B) and is easy to calculate. As a useful measure of association, it has been considered deficient by several investigators because it does not take into account the number of co-occurrences of D_A and D_B which are to be expected on the basis of chance. This evidently is a function of the magnitudes not only of A and B, but also of N, which does not appear in (1) and (2). To overcome this objection, Maron and Kuhns introduce a third measure, a contingency estimate, which removes from f the magnitude to be expected, on the basis of chance, given the actual values of A, B, and N:

$$S(D_A, D_B) = f - \frac{AB}{N} .$$

They then introduce an arbitrary coefficient of association, based upon S, ranging in value from -1 to +1 and equal to zero when S = 0. This coefficient is of the form

$$Q(D_A, D_B) = \frac{SN}{xy + wo} . \tag{3}$$

Stiles [11] also starts with the contingency table given above, and, using the Yates correction for a 2x2 table with one degree of freedom, adopts as an "association factor" (A.F.) the base 10 logarithm of the expression for χ^2:

$$A.F. = \log_{10} \chi^2 = \log_{10} \frac{(|fN - AB| - \frac{N}{2})^2 N}{AB(N - A)(N - B)} . \tag{4}$$

In use, all co-occurrences having A.F. ≥ 1 are retained as having potential usefulness, others being discarded. At this point, there is a probability on the order of 0.001 that an observed frequency of co-occurrence, f, is due to chance factors for the given values of A, B, and N. Association factors of 5 or more ($\chi^2 \geq 100,000$) are not unusual in libraries of more than 100,000 documents.

Doyle [12] introduces another measure to indicate strength of association:

$$S.A. = \frac{fN}{AB} . \tag{5}$$

This has a wide range of values and, because frequently N >> AB, may be quite large for small f. It is, of course, zero when f = 0, i.e., when the pair $D_A D_B$ does not exist in any document.

The expressions (1) to (5) all are based upon the total population of indexed documents, N, which is divided into four subsets:

(1) Those containing the term D_A.

(2) Those containing D_B.

(3) Those containing both D_A and D_B.

(4) Those containing neither term.

They include normalizing procedures to adjust the sizes of the group f to remove the effect that may result from the tendency of D_A and D_B, considered separately, to occur frequently as index terms. Such normalization is required because, the more frequently an index term occurs, the more frequently it is apt to be used with some other term simply on a chance basis.

2. Usefulness of Associations Which Occur Only a Few Times.

In most cases, it is extremely dubious if any particular significance can be attached to a unique index term "association." This is self-evident if one of terms, A, appears in only one document. If it contains c terms, A must form c - 1 single-occurrence pairs, regardless of the "statistical odds" against any particular pair AB. Similarly, terms used in only a few documents tend to form mostly unique pairs--over 95% in the DDC sample for A = 2 to 5. Although the percentage of multiple occurrences increases with A and B, even the 599 most common have 40% of their different pairs unique. Theoretically, a frequency distribution of expected pair occurences, based on chance, could be calculated for each of them. However, even if the number of unique pairs for a given A differs significantly from the chance expectation, in many cases there is no way of determining whether or not a specific pair AB represents a significant association.

The cases where f is small--say 2 to 5--may require more detailed analysis than they have so far received. If A also is small, then $P(D_B|D_A)$ may be meaningful. For example, f = 2 and A = 3 give some reason to believe that A, which co-occurs with B in two of its three uses, may have a significant association with B. The degree of confidence is strengthened if the indexing of additional documents creates such ratios as 4/6 or 5/7 and decreased if they become, say, 2/5 or 3/8. It is possible, but considered unlikely, that the limited amount of information in a single occurrence increases sharply, simply by adding another occurrence. In any event, it appears as if some attention should be paid to these occurrences, with the specific objective of ascertaining parametric criteria for distinguishing the "meaningful" from "nonmeaningful."

However, if A and B are relatively large, then small values of f may indicate a significant "negative association" between them. The theoretical frequency of co-occurrence, assuming independence, is

$$f_t = \frac{AB}{N}$$

and, if this value ≥ 5, the difference between observed and theoretical frequencies can be tested by standard statistical methods for significance. In the DDC sample, for example, the two high-usage terms "Temperature" (6th ranked with 1,489 occurrences) and "Countermeasures" (20th, with 846) occur together in only one document. The difference between the theoretical frequency of 33 co-occurrences and the one actually observed has a very small probability of being explainable by chance and it is concluded that the two terms have a significant negative association. [In equation (4) of Section 1, this occurs when fN - AB is negative.] In general, a significant negative association can be established statistically only when $AB \geq 5N$, or a little less if the case f = 0 (no co-occurrences) is considered. Because at least one of the terms must be used in $\sqrt{5N}$ or more documents, only a small percentage of possible or actual pairs are susceptible to this determination. In the DDC sample, only 50 terms occur more than $\sqrt{5N}$ = 438 times; only 17,900 pairs have $AB \geq 5N$.

3. The Conditional Probability $P(D_A|D_B) = f/B$.

This is easy to calculate and interpret: If a given document contains the term D_B, it is the probability that it also contains D_A. However, its significance is difficult to measure. f/B is independent of the actual magnitudes of f and B; it does not involve A at all, except that by definition $A \geq f$; and without introducing N, it cannot be determined whether or not f represents a significant association.

Despite these deficiencies, the conditional probability has one feature considered definitely desirable: It is a measure of the association in the direction required by the search request. For most pairs, $P(D_A|D_B)$ and $P(D_B|D_A)$ not only differ, but differ markedly. Whether or not a term should be added to the search request can well depend upon which one already is in it. If, for example $P(D_A|D_B) = 5/6$ and $P(D_B|D_A) = 5/200$, it is not at all obvious that identical actions should be taken regardless of which of the two terms is in the original request. Additionally, statistical tests for the significance of f do not depend upon the individual values of A and B, but only upon their product. The conditional probabilities definitely increase our knowledge of the nature of the association.

The frequency distribution of Table A-4 (Appendix A) gives $P(D_A|D_B)$, rounded to two decimal places, for all pairs among the 599 most frequently used DDC index terms. Note that entries for $f/B = .01$ include the 40,436 pairs ($D_iD_j \neq D_jD_i$) occurring only once. (The maximum value of $1/B$ is 1/72, which rounds to .01.) This distribution probably is roughly typical when both D_A and D_B have fairly high usage. It would be quite different if all index terms were included. For example, index terms used in from 1-10 documents form a quite large number of different pairs for which f = 1 or 2, resulting in pronounced peaks at the values 1/B, B = 1 to 10.

4. Association Factors and Coefficients.

Equations (3) and (4) of Section 1 are typical of association coefficients designed to indicate the probability that an observed frequency of co-occurrence will differ from the theoretical frequency by purely chance factors. The basic approach uses the 2x2 contingency table, whose cell entries can be determined readily from the known values of f, A, B, and N. The hypothesis that A and B are independent is tested by the χ^2 statistic. Because Stiles' Association Factor is the logarithm of a computational approximation to χ^2, it is used here for illustrative purposes:

$$A.F. = \log_{10} \chi^2 = \log_{10} \frac{(|fN - AB| - 0.5N)^2 N}{AB(N - A)(N - B)} . \qquad (6)$$

When $fN > AB$, which almost always is the case in document descriptions, the observed frequency is greater than the theoretical frequency. If f and N are fixed, and A and B are relatively small compared with N, then χ^2 (or A.F.) varies inversely as the magnitude of the product AB. As AB decreases to its minimum possible value $f^2 (A = B = f)$, χ^2 increases to its maximum value. An idea of the range of values of A and B for which χ^2 will exceed any desired value thus can be obtained once the product AB is known. The tables on the next page give these values for $\chi^2 \geq 10, 100$ and 1,000--corresponding to A.F. ≥ 1, 2, and 3--and for several values of f and N. The three tables at the left give the maximum value of B, which occurs when A = f. For example, if N = 50,000 and A = f = 5, then $\chi^2 \geq 10$ for all $B \leq 13{,}563$; $\chi^2 \geq 100$ for $B \leq 1{,}929$; and $\chi^2 \geq 1{,}000$ for $B \leq 201$. The right-hand three tables give the maximum value of the product AB; in the example, above, $\chi^2 \geq 10$ for all $AB \leq 67{,}815$.

If AB is considerably less than N--say 0.1N or less--χ^2 is given approximately by

$$\chi^2 \approx \frac{(f - \frac{1}{2})^2 N}{AB} . \qquad (7)$$

It is evident at once that the value of χ^2 is extremely sensitive to and increases rapidly with f, particularly when f is small.

The A.F. proposed by Stiles compresses these wide variations by using the logarithm of χ^2 itself. AF = 1.00 when $\chi^2 = 10$, for example, and A.F. = 3 for $\chi^2 = 1000$.

The appropriateness of using contingency tables, and specifically the 2x2, and the χ^2 statistic is questionable. Equation (6) approximates the χ^2 distribution only when the theoretical frequencies in each cell are reasonable in magnitude and in practice should not be used unless each such cell entry is at least 5. In the case of index term associations, the theoretical frequencies A, B, and N are taken to be the same as those observed, N always being quite large. Many of the A and B are less than 5, the exact percentage varying with library size, number of different indexing terms, depth of indexing, etc. However, the theoretical frequency of co-occurrences,

$$f_t = \frac{AB}{N} ,$$

practically never is as great as 5. It will not be unless $AB \geq 5N$ and typically is much less than 1.0. The "χ^2" calculated in these cases is difficult to interpret and its meaning becomes progressively more nebulous as its magnitude increases. In particular, there is no good reason to conclude that large differences in the magnitude of two χ^2's actually represent any real difference in the "degree of association" of two pairs of index terms, or that the two χ^2 values can be used as measures of the degrees of

Table 5

Association Factors:
Maximum Values of B and AB for Which $\chi^2 \geq 10$, 100, and 1000

Maximum B for $\chi^2 \geq 10$

N \ A=f	1	2	3	5	10	20
1,000	23	95	161	272	454	638
5,000	116	474	807	1,357	2,263	3,176
10,000	232	948	1,614	2,713	4,524	6,349
38,402	893	3,642	6,201	10,417	17,370	24,368
50,000	1,163	4,742	8,074	13,563	22,616	31,727
100,000	2,326	9,484	16,149	27,125	45,230	63,451
500,000	11,630	47,422	80,745	135,620	226,143	317,241
1,000,000	23,268	94,821	161,491	271,241	452,284	634,477

Maximum AB for $\chi^2 \geq 10$

N \ f	1	2	3	5	10	20
1,000	23	190	483	1,360	4,540	12,760
5,000	116	948	2,421	6,785	22,630	63,520
10,000	232	1,896	4,842	13,565	45,240	126,980
38,402	893	7,284	18,603	52,085	173,700	487,360
50,000	1,163	9,484	24,222	67,815	226,160	634,540
100,000	2,326	18,968	48,447	135,625	452,300	1,269,020
500,000	11,630	94,844	242,235	678,100	2,261,430	6,344,820
1,000,000	23,268	189,642	484,473	1,356,205	4,522,840	12,689,540

Maximum B for $\chi^2 \geq 100$

N \ A=f	1	2	3	5	10	20
1,000	2	11	20	38	82	161
5,000	12	55	101	192	410	794
10,000	24	110	202	385	820	1,584
38,402	95	424	707	1,482	3,151	6,087
50,000	124	552	1,012	1,929	4,103	7,922
100,000	248	1,104	2,024	3,859	8,206	15,845
500,000	1,240	5,521	10,121	19,295	41,032	79,228
1,000,000	2,481	11,043	20,243	38,590	82,065	158,457

Maximum AB for $\chi^2 \geq 100$

N \ f	1	2	3	5	10	20
1,000	2	22	60	190	820	3,220
5,000	12	110	303	960	4,100	15,880
10,000	24	220	606	1,925	8,200	31,680
38,402	95	848	2,121	7,410	31,510	121,740
50,000	124	1,104	3,036	9,645	41,030	158,440
100,000	248	2,208	6,072	19,295	82,060	316,900
500,000	1,240	11,042	30,363	96,475	410,320	1,584,560
1,000,000	2,481	22,086	60,729	192,950	820,650	3,169,140

Maximum B for $\chi^2 \geq 1,000$

N \ A=f	1	2	3	5	10	20
1,000	---	---	---	---	---	---
5,000	1	5	10	20	44	93
10,000	2	11	20	40	89	186
38,402	9	43	79	154	343	716
50,000	10	56	103	201	446	931
100,000	24	112	207	403	893	1,863
500,000	104	560	1,038	2,015	4,467	9,319
1,000,000	249	1,121	2,071	4,030	8,935	18,639

Maximum AB for $\chi^2 \geq 1,000$

N \ f	1	2	3	5	10	20
1,000	---	---	---	---	---	---
5,000	1	10	30	100	440	1,860
10,000	2	22	60	200	890	3,720
38,402	9	86	237	770	3,430	14,320
50,000	10	112	309	1,005	4,460	18,620
100,000	24	224	621	2,015	8,930	37,260
500,000	104	1,120	3,114	10,075	44,670	186,380
1,000,000	249	2,242	6,213	20,150	89,350	372,780

association. Consequently, the ordering into sequence of all terms associated in use with a given D_A, based upon the value of χ^2, does not give any assurance that the resultant order of the D_B is even approximately correct. The uncertainty probably is greatest for the larger values of χ^2. Because these values, averaged and/or normalized, ultimately become document "relevance numbers," a similar uncertainty exists in them.

It must be observed that the use of association measures based upon the 2x2 contingency table has produced apparently useful results, even though the approach itself is open to theoretical question. Usefulness of results, of course, is the ultimate test of any measure of association and the χ^2 statistic may well be useful. Certainly one objective of a retrieval system can be to order documents according to their probable relevance to the request and this ordering possibly need be only approximately correct. As a matter of note, so long as the determination of "degree of relevancy" is subjective and not assigned an empiric value, the evaluation of the "relevance numbers" by which documents are ordered is itself subjective. The important factor may not be the relevance number itself, but the fact that the documents most likely to be pertinent are grouped roughly at the top of the list.

D. THESAURUS STRUCTURE, INDEXING STANDARDS AND ASSOCIATION FACTORS

The study of association factors and their possible uses involves consideration of many factors; of great importance--and too often neglected in analyses--is the data base of document descriptions from which the association factors are calculated; they can be no better than the index terms assigned to documents. This section discusses the large class of associations implicit in the organization and structure of the thesaurus and suggests a general method in which they can be handled efficiently.

1. Hierarchal Nature of a Thesaurus.

The index terms in the thesaurus form a hierarchy, or tree-like structure, branching out from a relatively few major divisions at the top through a varying number of branch points or nodes down to the most detailed terms at the bottom of the inverted tree. The number of levels or branches varies in different parts of the tree, as does the number of terms at any one level.

Once the tree structure has been established and the relationships of index terms defined by "links" from one node to that above or those below, it is possible to enter the tree at any index term and traverse it in either direction using only the link data. This can be done by a computer, provided the linkage data are included in the thesaurus made available to it. This possibility has several important implications on the overall design and operation of the retrieval system, in addition to its effects on index term associations.

a. Implicit Index Term Associations. The thesaurus tree immediately specifies the members of a set of significant index term associations. A term D_n at level n always is a subset of the next higher term D_m at level m. Furthermore, $P(D_m \cdot D_n) = 1$. Conversely, D_m always includes as subsets all the

D_{ni} linked directly to it, but usually $p(D_{ni}|D_m) < 1$. In a similar manner, the term D_m bidirectionally linked through D_n with index terms at still higher levels. All of these index term associations derived from the thesaurus tree are significant, whether or not any particular pair meets tests for statistical significance.

b. Lowest Level Indexing. Only the lowest level or most detailed term applicable in any one branch need be assigned to a document. All higher level terms of more general meaning can be assigned automatically. With manual indexing, this not only saves some indexing effort and input data preparation, but also--and more important--assures that these higher-level terms are assigned.

c. Current Indexing Practices and Factor Association Studies. Automatic assignment of tree-related terms assures a degree of uniformity and completeness missing in every operative document retrieval system which has been examined. For a number of perfectly normal reasons, the assignment of tree-related terms to documents is quite variable. Sometimes several levels of terms in one branch are assigned; at others, only the (presumably) lowest level term applicable. Spot-checks of document descriptions in several applications against the thesaurus indicate that this variability is commonplace.

Although these spot-checks are fairly few in number, they all tend to indicate that existing files of document descriptions are missing an unknown, but possibly quite large, number of implicit term associations. Consequently, association factor studies based upon an existing file have utilized a data base known to be (or almost certainly) incomplete in a critical area of interest--the associations of index terms in a given small subset of the thesaurus. This known lack of coverage casts doubt upon the validity of all association measures calculated from the term pairs actually present.

2. Synonymous Index Terms.

It would appear that the principal cause of synonymous indexing terms is failure to recognize that a new term already is included in the definition of another. This in turn may be more common when the thesaurus does not define the precise meaning or scope of each term, but leaves the definition to variable human interpretation. Although it is possible that two synonymous terms can be matched because of significant associations with a common third term or set of terms, it is believed that the feasibility of the method has not been established. The DDC sample contains several hundred thousand matchings of two terms with a third, few of which are synonyms, and there is no obvious method by which they can be segregated. It is considered that the potential use of association measures as a means of identifying synonyms requires more justification than it has had so far.

3. "General" Indexing Terms.

Every thesaurus contains a number of indexing terms comparable to those in DDC Group 292, "General Concepts"--Analysis, Design, Errors, Measurement, Reliability, Standards, Tests, Theory, etc. In addition, there exist a number of other terms of very general meaning and wide applicability, of which examples are Mechanical Properties, Physical Properties, Production,

Bibliography, and many indexing entries in the field of mathematics. Finally, terms in the first two or three levels at the "top of the tree" in a major division or field of interest usually are fairly broad in meaning.

All of these are widely used in indexing documents. In the DDC sample, 99% of the documents include at least one term used 40 times or more and over half include terms with total usages of over 1,000. (There are only 15 of the latter.) These percentages would be even greater if the indexing uniformly included higher-level terms in the thesaurus tree. Their very popularity of usage generates a large number of pair associations of which they are one member and a high percentage of the pairs occur often enough--which may be three times or less--to have "statistical significance." It seems doubtful that many of them have any practical utility in a document retrieval system.

The "profile" of almost every index term used more than 3-4 times contains several of these general terms. The chances then are quite good that most or all of the terms in a search request have a significant association factor with some of them, which may be used to expand the list of terms upon which the search is made. The final list of document numbers may include many which are completely extraneous. It is not immediately apparent that an article on "Penicillin" is germane to a request on "Copper Pipe" merely because both have a high degree of association to each of the terms "Test Equipment," "Quality Control," "Standards" and "Production." Conversely, an article on "Lead Pipe" or "Steel Pipe" well could be relevant.

It appears, then, that these common terms either should be eliminated as generators of additional terms or their use should be carefully circumscribed. As an example, the terms added could be limited to those contained in the same divisional thesaurus tree, or a part of it, that has one of the narrower-meaning terms of the request. This procedure requires identifying and earmarking all the common terms to be restricted in usage, as well as indicating for all other terms the thesaurus tree or subtree to which they belong. The precise method of making these identifications needs to be established.

E. TIME-INTERVAL SUBDIVISION OF ASSOCIATION FACTORS

The principal operative use of measure of association is to expand an original search request by adding to it other terms which have a significant number of co-occurrences with terms in the request or its first-order expansion. The presumption is that these terms will isolate otherwise unobtainable documents which may be relevant to the request. Insofar as retrieval is concerned, this is considered to be the most important potential use of association factors.

A document file is a dynamic organism and, by direct extension, so is the set of indexing terms and their associations. New terms are added to the thesaurus as new meanings or definitions are introduced into the fields of interest covered; existing terms may be combined or subdivided into several new ones to reflect the changing nature of documents. New associations of terms are generated as previously separated areas of endeavor become wedded. These changes are inherent in the basic data upon which the retrieval system

operates. In addition to these, procedure-dependent changes in these parameters are introduced by the normal effort to improve the system's effectiveness and responsiveness. These effects probably are most significant during the early years of operation, when revisions and modifications to the thesaurus, depth and type of indexing, and similar factors may be quite extensive.

This question arises naturally: Should the time parameter be introduced as a variable in analyses having to do with index term usage? There is considerable indirect evidence that this is highly desirable if not necessary. Although it is generally considered that reports and journal articles lose a good deal of their value after five years, it appears that most information centers will retain them in an active status for a longer period of time, possibly ten years. If the time parameter is not introduced, the values A, B, N, and f then simply are totals for some fairly long period and often will not reflect short-term changes. There may be nothing particularly significant for $f = 10$ if $A = 200$ and $B = 300$. The relationship could be quite significant if the co-occurrences took place within a 10% time range of D_A and D_B. It is precisely this sort of relationship that would be isolated by the time parameter.

Subdividing the file of index term usage into time intervals reduces the values A, B, and N, and the theoretical frequency f. Because the latter already is very small for most pairs of index terms, its further diminution places additional pressure on developing meaningful measures of association. File storage also increases, because now it is necessary to accumulate A, B, and f within each time interval. It is concluded that a complete evaluation of the use of index term associations requires analysis of the effects of the time parameter. So far as known, this has not yet been considered.

F. SIZE OF DOCUMENT SAMPLES FOR ASSOCIATION FACTOR STUDIES

Several of the published results on investigations into the derivation and use of association factors have been based upon fairly small samples of documents, usually less than about 500 and limited to one major subject classification of the library used. There is a good deal of doubt as to the general validity of these small-sample studies, particularly when results are to be extrapolated to an entire library. At least three different factors contribute to this uncertainty.

The first is that the complete file of document descriptions generates a multitude of small-magnitude statistics. Estimates, based upon sample data, of anything more than general characteristics are subject to quite large standard errors. Experience with two different random 10% samples (each of about 3,800 descriptions) from the 38,402-document DDC file probably are representative of these uncertainties. Estimating the number of different index terms in the full file from a sample is subject to an error of about 20%. Attempts to estimate the frequency distribution of their total usage, based upon the usages of terms included in the sample, have been largely unsuccessful, except for the 15% most commonly used. Because most term associations in the full file occur fewer than ten times, the samples have been of little value in studying them. Statistics based upon only a few hundred documents seldom will be representative of the full file.

Second, most small samples have been comprised of documents indexed over a short time interval and are not random. At best, they can represent only the documents described during the period the indexing standards of the sample were followed. They almost certainly are not typical of earlier documents.

Finally--and most important--samples limited to documents in one subject classification do not reflect the interactions of term associations introduced by documents in other classifications. Again referring to the DDC data, 90% of the different pairs and 85% of their occurrences involve terms in two different fields of interest. The typical document uses terms from several groups and fields and the existence of a given interfield pair usually gives no useful clue as to the subject classification of the document. It is considered virtually certain that association factor studies based upon single-subject document data have a definite bias in favor of the usefulness of the results. By the nature of the sample, all terms added in the first and second-order cycles must lead to documents pertaining to the one subject area. One would expect these to have a much higher average chance of being relevant to a request than documents classified under other subjects. Actual operating conditions are quite different. Here the values of factors used in the term association formula employed are determined by total library usage, as is the calculated measure of association, and the list of retrieved documents, with or without relevance numbers, is not confined to those pertaining to a single subject. Any proposed use of association factors must be adaptable to the entire library. The evaluation of their usefulness in retrieving documents likewise must be based upon the total operating enviornment, and not upon a nonrepresentative subset of it.

It is considered that representative studies into index term associations and their use must be based upon fairly large samples selected as a roughly random cross-section of a complete document library. The actual minimum number of documents required is rather difficult to stipulate and may vary somewhat depending upon the number of different index terms and average number of different index terms which have been assigned per document. A suggested minimum is in the 5-10,000 document range, with the entire file used if it is less than about 20,000. For larger files, the sample may range from around 50% of the documents down to possibly 20% for files of over 100,000. Admittedly, samples of this size involve quite large volumes of data which are rather expensive to process and this cost may create a severe strain on limited-budget research studies. On the other hand, unless the sample is large enough to generate a fairly good array of term associations, test results may have limited applicability, and perhaps none, to an operative system.

G. CONCLUSIONS

Although it is considered that index term associations may improve the operation of a document retrieval system, it is concluded that further research is necessary to establish the degree of improvement which may be expected. In addition, such studies should take into account the file storage and data processing aspects of their use.

It is considered desirable to distinguish between associations implicit in the thesaurus structure and term definitions on the one hand and those based simpl upon co-occurrence in usage on the other. Experimental studies must be based upon large samples representing a full cross-section of a library's coverage and the document descriptions must form a complete data base within the structure of the thesaurus, correcting the deficiencies which have existed in an unknown degree in almost all studies so far conducted. Investigation into meaningful measures of statistical significance of associations should be pursued and the usefulness of co-occurrences present only a few times established.

APPENDIX A

Table A-1A
599 Most Common DDC Descriptors With Field and Group Classifications (In Sequence by Frequency of Usage)

Rank	Diff. Pairs	Fld/Grp	Descriptor
1	571	13 292	Design
2	579	13 292	Tests
3	509	15 147	Mathematical Analysis
4	542	13 292	Measurement
5	444	01 114	Guided Missiles
6	491	15 117	Temperature
7	385	06 027	Airborne
8	418	09 217	Production
9	492	13 292	Theory
10	443	10 145	Materials
11	505	13 292	Analysis
12	362	06 027	Surface-to-Surface
13	466	07 108	Great Britain
14	437	01 006	Stability
15	450	13 292	Effectiveness
16	313	02 183	Flight Testing
17	279	02 227	Radar Equipment
18	445	17 208	Instrumentation
19	464	13 292	Test Methods
20	398	02 078	Countermeasures
21	383	11 216	Pressure
22	368	02 102	Detection
23	271	15 146	Mechanical Properties
24	436	13 292	Test Equipment
25	338	01 010	Control Systems
26	326	09 217	Processing
27	274	04 053	Synthesis
28	362	15 10[illegible]	Physical Properties
29	196	12 209	Physiology
30	320	15 117	Heat Transfer
31	275	[illegible] 076	Circuits
32	410	13 292	Determination
33	251	04 053	Chemical Reactions
34	232	06 027	Surface-to-Air
35	282	15 247	Stresses
36	283	04 106	Polymers
37	245	14 020	Projectiles
38	285	01 006	Aerodynamics
39	250	16 057	Combustion
40	238	01 009	Jet Planes
41	318	15 116	Radiation Effects
42	298	16 085	Rocket Motors
43	269	02 170	Guidance
44	307	13 292	Reliability
45	227	10 099	Solid Rocket Propellants
46	302	15 187	Propagation
47	293	15 178	Scattering
48	237	01 005	Model Tests
49	309	15 147	Statistical Analysis
50	331	01 006	Vibration
51	252	15 066	Crystals
52	214	06 273	Transistors
53	361	13 071	Bibliography
54	255	02 062	Storage
55	169	01 006	Supersonics
56	321	06 079	Electronic Equipment
57	193	06 082	Electron Tubes
58	356	01 009	Aircraft
59	239	15 247	Semiconductors
60	175	14 100	Fuzes
61	284	02 196	Military Requirements
62	323	15 148	Velocity
63	283	13 060	Digital Computers
64	247	04 131	Oxides
65	297	19 253	Satellite Vehicles
66	215	01 094	Jet Fighters
67	271	15 076	Electrical Properties
68	275	13 292	Sensitivity
69	224	01 006	Launching
70	270	13 292	Errors
71	251	14 090	Vulnerability
72	280	13 060	Computers
73	301	13 104	Operation
74	279	10 160	Metals
75	200	15 247	Deformation
76	236	15 187	Spectrographic Analysis
77	121	12 201	Pathology
78	283	15 117	Gases
79	305	15 117	Thermodynamics
80	310	07 108	Canada
81	192	02 227	Search Radar
82	222	05 061	Maintenance
83	234	15 187	Electromagnetic Waves
84	213	10 016	Steel
85	256	01 006	Shock Waves
86	238	06 025	Antennas
87	302	04 053	Reduction
88	161	14 048	Chemical Warfare Agents
89	194	06 027	X Band
90	190	11 256	Sheets
91	241	07 054	Atmosphere
92	260	10 212	Plastics
93	281	15 187	Absorption
94	242	15 187	Reflection
95	194	02 227	Radar Tracking
96	137	01 005	Wind Tunnel Models
97	241	10 056	Coatings
98	135	07 054	Meteorological Data
99	168	01 006	Drag
100	150	04 049	Silicon
101	218	13 060	Data Processing Systems
102	191	10 099	Liquid Rocket Propellants
103	201	06 027	Air-to-Air
104	133	08 223	Acceptability
105	246	11 275	Handling
106	176	13 060	Coding
107	171	01 094	Fighters
108	241	13 292	Configuration
109	194	01 114	Guided Missile Trajectories
110	185	14 100	Guided Missile Fuzes
111	178	15 066	Crystal Structure
112	151	09 217	Manufacturing Methods
113	270	17 136	Test Facilities
114	174	06 229	Radio Equipment
115	234	06 081	Microwaves
116	271	13 292	Control
117	194	16 219	Rocket Propulsion
118	174	15 025	Molecular Structure
119	176	12 209	Growth
120	174	06 059	Radio Communication Systems
121	173	02 067	Anti-Aircraft Defense Systems
122	228	15 247	Structures
123	248	14 020	Explosives
124	108	[illegible] 036	Biochemistry
125	201	13 060	Programming
126	203	07 054	Climatic Factors
127	262	04 049	Oxygen
128	209	07 054	Meteorology
129	152	04 131	Hydrides
130	218	15 105	Energy
131	196	06 059	Communication Systems
132	209	01 006	Gas Flow
133	180	15 147	Probability
134	207	09 217	Preparation
135	126	12 269	Toxicity
136	215	15 230	Particles
137	243	08 001	Hazards
138	177	06 027	Shipborne
139	176	01 006	Boundary Layer
140	177	07 108	Arctic Regions
141	238	15 148	Motion
142	230	15 029	Diffusion
143	200	01 005	Cylindrical Bodies
144	224	01 006	Aerodynamic Heating
145	243	[illegible] 237	Rockets
146	19[illegible]	16 [illegible]	Ignition
147	224	01 [illegible]	[illegible]
148	237	15 076	Resolution
149	206	15 076	Magnetic Fields
150	189	15 121	Hydrodynamics

Table A-1B

599 Most Common DDC Descriptors With Field and Group Classifications
(In Sequence by Frequency of Usage)

Rank	Diff. Pairs	Fld/Grp	Descriptor
151	091	07 054	Weather Forecasting
152	157	14 113	Guns
153	185	04 131	Fluorides
154	188	04 131	Chlorides
155	193	14 099	Penetration
156	202	10 145	Ceramic Materials
157	143	06 027	S Band
158	203	04 049	Hydrogen
159	180	08 119	Human Engineering
160	159	10 016	Aluminum Alloys
161	195	15 249	Acoustics
162	176	15 247	Elasticity
163	129	12 209	Inhibition
164	143	13 060	Data Transmission Systems
165	198	02 062	Containers
166	193	15 230	Gamma Rays
167	170	15 121	Fluid Flow
168	178	09 217	Corrosion
169	152	06 022	Amplifiers
170	169	06 027	Air-to-Surface
171	213	13 104	Specifications
172	163	01 041	Bombers
173	185	06 059	Display Systems
174	200	15 147	Tables
175	150	15 148	Tensile Properties
176	137	06 022	Microwave Amplifiers
177	115	03 098	Food
178	178	01 006	Thrust
179	174	04 053	Decomposition
180	134	01 005	Aerodynamic Configurations
181	083	12 154	Metabolism
182	197	02 107	Tracking
183	214	15 187	Interference
184	156	01 009	Helicopters
185	109	01 006	Lift
186	157	04 051	Chemical Analysis
187	157	06 079	Microwave Equipment
188	161	01 006	Load Distribution
189	235	04 157	Aluminum
190	151	07 054	Wind
191	157	15 230	Radioactive Isotopes
192	230	15 029	Ionization
193	157	04 049	Boron Compounds
194	164	02 163	Aerial Reconnaissance
195	188	15 029	Electrons
196	191	15 230	Radioactivity
197	141	14 115	Armament
198	166	06	Electrical Equipment
199	216	14 090	Detonation
200	155	06 232	Recording Devices
201	163	14 090	Blast
202	147	04 131	Ferrites
203	142	06 081	Radio Waves
204	191	06 215	Power Supplies
205	175	06 027	Ultra High Frequency
206	175	18 244	Submarines
207	173	15 147	Differential Equations
208	174	15 105	Intensity
209	147	06 092	Diodes
210	151	06 027	Broadband
211	225	15 105	Surface Properties
212	146	01 006	Flight Paths
213	118	01 005	Wings
214	178	14 020	Propellants
215	206	15 117	Cooling
216	212	13 292	Data
217	155	14 261	Tanks
218	152	01 006	Supersonic Flow
219	110	13 073	Scheduling
220	138	14 020	High-Explosive Ammunition
221	157	14 096	Fire Control Systems
222	203	07 054	Water
223	089	03 098	Preservation
224	137	09 217	Heat Treatment
225	146	15 093	Glass Textiles
226	238	15 105	Density
227	199	06 027	Automatic
228	194	07 108	USSR
229	174	15 211	Plasma Physics
230	154	15 287	Antenna Radiation Patterns
231	144	06 027	Very High Frequency
232	098	08 223	Psychology
233	225	15 247	Surfaces
234	186	04 053	Oxidation
235	189	13 306	Photographic Analysis
236	186	04 207	Ions
237	198	15 247	Solids
238	160	06 081	Signal-To-Noise Ratio
239	159	11 240	Safety Devices
240	196	14 032	Terminal Ballistics
241	164	01 012	Airframes
242	168	02 062	Packaging
243	095	15 147	Functions
244	093	03 086	Enzymes
245	186	15 133	Sources
246	100	01 006	Transonics
247	150	06 027	High Frequency
248	184	15 076	Electromagnetic Effects
249	191	15 117	Thermal Radiation
250	119	15 148	Specific Impulse
251	137	02 078	Radar Jamming
252	172	15 287	Wave Transmission
253	106	14 100	Radio Proximity Fuzes
254	122	10 016	Titanium Alloys
255	147	04 106	Organic Compounds
256	108	06 082	Traveling Wave Tubes
257	183	01 041	Jet Bombers
258	176	15 076	Dielectrics
259	158	13 292	Standards
260	085	12 266	Therapy
261	163	17 234	Scientific Research
262	177	13 292	Calibration
263	119	10 158	Fatigue (Mechanics)
264	188	06 074	Electronic Circuits
265	154	09 217	Deterioration
266	140	10 158	Shock Resistance
267	170	06 081	Radar Reflections
268	114	10 146	Binders
269	136	10 004	Seals
270	129	06 229	Radio Receivers
271	144	09 217	Aging
272	161	10 099	Rocket Propellants
273	169	02 127	Infrared Detectors
274	158	01 009	Airplanes
275	121	11 275	Transportation
276	176	15 076	Polarization
277	188	15 187	Light
278	128	10 099	Rocket Oxidizers
279	168	12 209	Life Expectancy
280	158	14 286	Guided Missile Warheads
281	178	15 187	Optics
282	154	01 008	Aircraft Equipment
283	136	16 09[illegible]	Turbojet Engines
284	173	16 219	Propulsion
285	141	01 006	Turbulence
286	141	15 147	Sampling
287	198	13 060	Analog Computers
288	141	04 018	Amines
289	123	15 187	Visibility
290	117	02 0[illegible]3	Radar Interception
291	152	15 187	Infrared Spectroscopy
292	150	15 076	Electromagnetic Properties
293	104	13 246	Selection
294	205	11 240	Safety
295	082	14 020	Cartridges
296	140	10 212	Laminates
297	155	15 212	Gas Ionization
298	144	15 116	Contamination
299	128	01 005	Bodies of Revolution
300	217	15 287	Attenuation

Table A-1C

599 Most Common DDC Descriptors With Field and Group Classifications (In Sequence by Frequency of Usage)

Rank	Diff. Pairs	Fld/Grp	Descriptor
301	147	15 066	Single Crystals
302	099	14 100	Projectile Fuzes
303	155	07 054	Upper Atmosphere
304	100	13 071	Classification
305	092	08 223	Behavior
306	163	13 073	Instruction Manuals
307	182	19 251	Re-entry Aerodynamics
308	156	01 006	Hypersonics
309	141	16 173	Rocket Motor Nozzles
310	125	04 157	Germanium
311	104	15 147	Matrix Algebra
312	184	16 099	Exhaust Gases
313	079	14 048	G Agents
314	132	06 059	Communication Equipment
315	134	15 066	Microstructure
316	166	04 003	Ethylenes
317	095	15 247	Creep
318	097	04 053	Polymerization
319	152	19 251	Atmosphere Entry
320	130	15 029	X Rays
321	193	19 253	Spaceships
322	159	15 249	Noise
323	091	08 270	Military Training
324	107	10 158	Fracture (Mechanics)
325	112	14 020	Fin-Stabilized Ammunition
326	127	15 211	Shock Tubes
327	139	10 099	Jet Engine Fuels
328	150	01 114	Guided Missile Noses
329	105	04 193	Hydrazines
330	182	04 049	Nitrogen
331	186	02 102	Detectors
332	183	15 247	Conductivity
333	111	14 100	Arming Devices
334	111	04 193	Urethanes
335	116	19 251	Satellite Vehicle Trajectories
336	150	15 247	Gradiation Damage
337	158	04 106	Liquids
338	144	19 099	Fuels
339	102	07 045	Mapping
340	087	12 023	Tissues (Biology)
341	143	06 027	Radiofrequency
342	180	17 234	High Temperature Research
343	156	10 158	Failure (Mechanics)
344	110	14 020	Antitank Ammunition
345	133	15 066	X-Ray Diffraction Analysis
346	138	15 121	Viscosity
347	124	04 049	Nitrogen Compounds
348	141	04 207	Purification
349	132	15 066	Lattices
350	094	10 056	Corrosion Inhibition
351	053	08 223	Attitudes
352	193	15 187	Infrared Radiation
353	120	14 286	Warheads
354	125	14 100	Proximity Fuzes
355	117	04 131	Perchlorates
356	114	07 054	Moisture
357	150	06 215	Generators
358	188	15 187	Frequency
359	151	01 006	Wind Tunnels
360	104	06 027	L Band
361	092	13 073	Industrial Production
362	147	15 133	Impurities
363	143	10 112	Glass
364	133	06 166	Frequency Modulation
365	122	04 190	Methyl Radicals
366	134	15 065	Liquefied Gases
367	118	07 054	Ionosphere
368	121	01 005	Control Surfaces
369	113	06 274	Wave Guides
370	113	17 080	Test Sets
371	172	14 032	Range
372	130	10 099	Propellant Grains
373	115	15 147	Numerical Analysis
374	187	19 253	Hypervelocity Vehicles
375	119	02 102	Direction Finding
376	116	08 202	Aviation Personnel
377	114	09 217	Quality Control
378	133	15 187	Oscillation
379	123	01 006	Hypersonic Flow
380	129	06 025	Radar Antennas
381	101	10 099	Propellant Properties
382	129	10 159	Metallurgy
383	114	13 060	Data Storage Systems
384	139	14 040	Aerosols
385	125	10 182	Lubricants
386	134	12 150	Identification
387	132	02 227	Doppler Radar
388	136	09 217	Bonding
389	187	07 054	Air
390	144	15 121	Absorption
391	216	13 071	Symposia
392	123	10 016	Stainless Steel
393	131	01 006	Jets
394	138	14 [illegible]	Exterior Ballistics
395	113	09 [illegible]	Casting
396	138	15 247	Thin Films
397	165	01 006	Stabilization
398	124	02 227	Radar Receivers
399	140	13 073	Costs
400	072	15 066	Quartz Crystals
401	087	15 147	Partial Differential Equations
402	106	15 076	Magnetic Properties
403	135	06 079	Electronic Systems
404	087	01 006	Laminar Boundary Layer
405	152	14 072	Ballistics
406	114	01 255	Transport Planes
407	116	10 145	Refractory Materials
408	091	01 005	Triangular Wings
409	157	15 249	Transducers
410	094	01 006	Stability (Longitudinal)
411	086	06 082	Magnetrons
412	143	15 148	Impact Shock
413	168	01 006	High Altitude
414	120	06 075	Electrodes
415	096	16 057	Combustion Chambers
416	143	02 180	Atomic Bomb Explosions
417	152	11 282	Vehicles
418	131	02 227	Radar Targets
419	179	15 117	Heating
420	135	15 122	Erosion
421	138	15 076	Dielectric Properties
422	118	15 [illegible]	Electron Beams
423	092	15 116	Dose Rate
424	148	04 [illegible]	Separation
425	175	02 227	Radar
426	136	02 170	Navigation
427	089	09 288	Welding
428	140	04 131	Sulfides
429	138	04 049	Sodium Compounds
430	087	15 225	Quantum Mechanics
431	[illegible]	04 131	Nitrates
432	104	10 145	Ferromagnetic Materials
433	119	08 202	Military Personnel
434	105	14 165	Mines
435	069	08 223	Learning
436	052	12 266	Diet
437	136	04 049	Aluminum Compounds
438	092	15 [illegible]	Underwater Sound
439	105	15 211	Magnetohydrodynamics
440	060	04 207	Dehydration
441	215	01 006	Simulation
442	081	14 048	V Agents
443	076	01 005	Wing-Body Configurations
444	114	15 117	Phase Studies
445	134	06 027	Low Frequency
446	127	10 099	Rocket Fuels
447	113	06 081	Radio Interference
448	105	15 029	Molecules
449	096	09 217	Machining
450	138	16 057	Flames

Table A-1D
599 Most Common DDC Descriptors With Field and Group Classifications
(In Sequence by Frequency of Usage)

Rank	Diff. Pairs	Fld/Grp	Descriptor
451	135	10 146	Films
452	105	15 076	Electrical Networks
453	098	14 020	Armor Piercing Ammunition
454	138	15 200	Targets
455	097	06 074	Switching Circuits
456	133	02 102	Range Finding
457	109	02 153	Landing
458	132	15 143	Friction
459	132	01 006	Damping
460	114	01 005	Conical Bodies
461	129	13 246	Conferences
462	114	15 117	Thermal Stresses
463	087	10 239	Synthetic Rubber
464	123	07 054	Ice
465	081	01 006	Flutter
466	103	08 270	Training
467	089	13 104	Maneuverability
468	120	19 253	Lunar Probes
469	117	04 087	Esters
470	150	07 028	Earth
471	060	12 209	Nutrition
472	153	10 145	Insulating Materials
473	110	02 170	Inertial Guidance
474	112	14 042	Bombs
475	100	01 005	Blunt Bodies
476	129	06 027	Radiofrequency
477	106	01 006	Moments
478	119	15 249	Sound
479	[illegible]	12 023	Skin
480	[illegible]	18 244	Ships
481	049	08 223	Group Dynamics
482	079	07 108	Geography
483	114	15 187	Diffraction
484	119	07 181	Sea Water
485	118	04 131	Peroxides
486	079	04 190	Nitro Radicals
487	120	15 186	Infrared Equipment
488	146	13 071	Handbooks
489	119	04 106	Chemical Properties
490	158	15 148	Acceleration
491	085	12 209	Visual Perception
492	112	06 082	Klystrons
493	100	15 147	Integral Equations
494	065	14 100	Bomb Fuzes
495	107	01 006	Subsonic Flow
496	139	19 251	Space Flight
497	129	10 164	Soils
498	114	13 104	Digital Systems
499	120	15 247	Deflection
500	092	02 170	Command Systems
501	123	10 016	Alloys
502	155	15 249	Resonance
503	061	08 202	Naval Personnel
504	096	01 009	Naval Aircraft
505	087	10 182	Lubrication
506	139	05 061	Installation
507	081	14 115	Gun Barrels
508	072	09 217	Extrusion
509	093	15 225	Electron Transitions
510	087	04 051	Electrochemistry
511	064	12 072	Drugs
512	068	12 023	Blood
513	086	10 004	Adhesives
514	108	08 270	Training Devices
515	078	15 178	Resonance Absorption
516	129	07 054	Precipitation
517	134	01 005	Spheres
518	075	03 221	Proteins
519	093	15 247	Photoconductivity
520	116	15 186	Optical Systems
521	102	10 016	Nickel Alloys
522	118	04 049	Lead Compounds
523	081	14 100	Firing Mechanisms
524	109	02 227	Early Warning Radar
525	143	04 157	Copper
526	072	13 202	Cargo Vehicles
527	108	04 131	Carbides
528	121	01 114	Aerial Targets
529	052	15 147	Statistical Processes
530	106	10 239	Rubber
531	146	01 114	Recovery
532	138	06 197	Oscillators
533	102	15 076	Impedance
534	144	15 076	Electric Fields
535	150	15 148	Dynamics
536	096	04 190	Vinyl Radicals
537	062	08 223	Reasoning
538	101	15 105	Porosity
539	119	04 157	Molybdenum
540	095	02 226	Military Equipment
541	097	15 147	Information Theory
542	099	13 104	Engineering
543	081	06 082	Backward-Wave Oscillators
544	077	01 005	Airfoils
545	068	01 009	Vertical Take-off Planes
546	094	01 006	Turbulent Boundary Layer
547	122	15 147	Perturbation Theory
548	126	04 049	Carbon
549	081	15 147	Series
550	116	02 227	Radar Echo Areas
551	102	04 053	Pyrolysis
552	100	10 212	Epoxy Resins
553	131	04 051	Chemistry
554	094	14 026	Armor Plate
555	143	04 049	Silicon Compounds
556	118	06 081	Radar Signals
557	077	19 251	Orbital Flight Paths
558	113	14 020	Fragmentation Ammunition
559	096	14 090	Fragmentation
560	117	15 025	Excitation
561	091	02 067	Aircraft Defense Systems
562	063	02 184	Air Drop Operations
563	079	06 059	Voice Communication Systems
564	086	01 114	Target Drones
565	111	14 096	Surface Targets
566	055	08 223	Reaction (Psychology)
567	115	01 006	Ablation
568	069	12 209	Vision
569	127	04 106	Vapors
570	065	02 002	Sonar Equipment
571	075	02 078	Radio Jamming
572	105	08 202	Pilots
573	099	15 187	Microwave Spectroscopy
574	127	04 049	Hydrogen Compounds
575	124	15 076	Electromagnetic Fields
576	[illegible]	10 239	Elastomers
577	045	13 246	Culture
578	097	01 006	Turbulent Flow
579	101	10 239	Military Research
580	130	14 237	Guided Missile Launchers
581	084	04 051	Chromatographic Analysis
582	115	06 075	Capacitors
583	075	14 115	Automatic Weapons
584	070	04 157	Uranium
585	111	01 114	Telemeter Systems
586	097	01 006	Supersonic Wind Tunnels
587	094	15 116	Monitors
588	094	10 159	Metallurgical Analysis
589	102	04 049	Lithium Compounds
590	053	13 071	Documentation
591	041	03 098	Dehydrated Substances
592	112	05 061	Construction
593	093	15 260	Servo Systems
594	098	14 020	Release Mechanisms
595	076	02 078	Radio Interception
596	093	01 198	Parachutes
597	114	04 207	Mixtures
598	113	06 079	Miniature Electronic Equipment
599	089	11 092	Joints

Table A-2
Pair Associations Among the 599 Most Common DDC Index Terms
Classified by Number of Occurrences

Occur-rences	No. of Different Pairs No.	%	Cum.No.	%	No. of Pair Occurrences No.	%	Cum.No.	%
1	20,218	41.01	20,218	41.01	20,218	8.14	20,218	8.14
2	8,654	17.55	28,872	58.56	17,308	6.97	37,526	15.11
3	4,854	9.84	33,726	68.40	14,562	5.86	52,088	20.97
4	3,180	6.45	36,906	74.85	12,720	5.12	64,308	26.09
5	2,058	4.17	38,964	79.02	10,290	4.14	75,098	30.23
6	1,642	3.33	40,606	82.36	9,852	3.97	84,950	34.20
7	1,219	2.47	41,825	84.83	8,533	3.43	93,483	37.63
8	953	1.93	42,778	86.76	7,624	3.07	101,107	40.70
9	779	1.58	43,557	88.34	7,011	2.82	108,118	43.52
10	638	1.29	44,195	89.63	6,380	2.57	114,498	46.09
11	537	1.09	44,732	90.72	5,907	2.38	120,405	48.47
12	435	0.88	45,167	91.61	5,220	2.10	125,625	50.57
13	368	0.75	45,535	92.35	4,784	1.93	130,409	52.49
14	368	0.75	45,903	93.10	5,152	2.07	135,561	54.57
15	316	0.64	46,219	93.74	4,740	1.91	140,301	56.48
16	273	0.55	46,492	94.29	4,368	1.76	144,669	58.23
17	229	0.46	46,721	94.76	3,893	1.57	148,562	59.80
18	187	0.38	46,908	95.14	3,366	1.35	151,928	61.16
19	177	0.36	47,085	95.50	3,363	1.35	155,291	62.51
20	163	0.33	47,248	95.83	3,260	1.31	158,551	63.82
21	143	0.29	47,391	96.12	3,003	1.21	161,554	65.03
22	113	0.23	47,504	96.35	2,486	1.00	164,040	66.03
23	96	0.19	47,600	96.54	2,208	0.89	166,248	66.92
24	96	0.19	47,696	96.73	2,304	0.92	168,552	67.85
25	87	0.18	47,783	96.91	2,175	0.88	170,727	68.72
26	106	0.21	47,889	97.13	2,756	1.11	173,483	69.83
27	74	0.15	47,963	97.28	1,998	0.80	175,481	70.64
28	74	0.15	48,037	97.43	2,072	0.83	177,553	71.47
29	64	0.13	48,101	97.56	1,856	0.75	179,409	72.22
30	72	0.15	48,173	97.70	2,160	0.87	181,569	73.09
31	63	0.13	48,236	97.83	1,953	0.79	183,522	73.87
32	61	0.12	48,297	97.95	1,952	0.79	185,474	74.66
33	46	0.09	48,343	98.05	1,518	0.61	186,992	75.27

Occur-rences	No. of Different Pairs No.	%	Cum.No.	%	No. of Pair Occurrences No.	%	Cum.No.	%
34	52	0.11	48,395	98.15	1,768	0.71	188,760	75.98
35	38	0.08	48,433	98.23	1,330	0.54	190,090	76.52
36	31	0.06	48,464	98.29	1,116	0.45	191,206	76.97
37	42	0.09	48,506	98.38	1,554	0.63	192,760	77.59
38	46	0.09	48,552	98.47	1,748	0.70	194,508	78.30
39	40	0.08	48,592	98.55	1,560	0.63	196,068	78.92
40	41	0.08	48,633	98.64	1,640	0.66	197,708	79.58
41	34	0.07	48,667	98.70	1,394	0.56	199,102	80.15
42	26	0.05	48,693	98.76	1,092	0.44	200,194	80.59
43	23	0.05	48,716	98.80	989	0.40	201,183	80.98
44	20	0.04	48,736	98.84	880	0.35	202,063	81.34
45	16	0.03	48,752	98.88	720	0.29	202,783	81.63
46	24	0.05	48,776	98.93	1,104	0.44	203,887	82.07
47	16	0.03	48,792	98.96	752	0.30	204,639	82.37
48	13	0.03	48,805	98.98	624	0.25	205,263	82.63
49	22	0.04	48,827	99.03	1,078	0.43	206,341	83.06
50-54	92	0.19	48,919	99.22	4,808	1.94	211,149	85.00
55-59	62	0.13	48,981	99.34	3,528	1.42	214,677	86.42
60-64	48	0.10	49,029	99.44	2,957	1.19	217,634	87.61
65-69	48	0.10	49,077	99.54	3,223	1.30	220,857	88.90
70-74	39	0.08	49,116	99.61	2,810	1.13	223,667	90.03
75-79	18	0.04	49,134	99.65	1,389	0.56	225,056	90.59
80-84	22	0.04	49,156	99.70	1,803	0.73	226,859	91.32
85-89	12	0.02	49,168	99.72	1,039	0.42	227,898	91.74
90-94	17	0.03	49,185	99.75	1,567	0.63	229,465	92.37
95-99	16	0.03	49,201	99.79	1,549	0.62	231,014	92.99
100-109	23	0.05	49,224	99.83	2,402	0.97	233,416	93.96
110-119	14	0.03	49,238	99.86	1,589	0.64	235,005	94.60
120-129	12	0.02	49,250	99.89	1,485	0.60	236,490	95.20
130-139	9	0.02	49,259	99.90	1,203	0.48	237,693	95.68
140-149	7	0.01	49,266	99.92	1,013	0.41	238,706	96.09
150&Over	40	0.08	49,306	100.00	9,719	3.91	248,425	100.00
Totals	49,306	—	—	—	248,425	—	—	—

Table A-3
Pair Associations Among the 599 Most Common DDC Descriptors, Classified by Number of Different Pairs and Total Occurrences

Total Pair Occurrences	Totals	No. of Different Pairs																		
		25-49	50-74	75-99	100-124	125-149	150-174	175-199	200-224	225-249	250-274	275-299	300-324	325-349	350-374	375-399	400-449	450-499	500-549	550-598
100-199	11	1	7	3																
200-299	74	2	10	30	22	10														
300-399	123		4	28	47	31	12	1												
400-499	82			15	19	27	13	6	2											
500-599	58			1	16	12	10	15	4											
600-699	52				4	6	17	17	5	3										
700-799	41				4	9	9	11	4	4										
800-899	28				3	4	6	5	6	3			1							
900-999	16					2	4	5	2	2	1									
1000-1099	10						1	1		4	1	1	1		1					
1100-1199	20					1	1	4	5	3	3	2	1							
1200-1299	10						2	2	1		1	2	2							
1300-1399	9					1	1	2		2	1	2								
1400-1499	5							1	1	1	2									
1500-1749	16						2	2	2	1	2	2	2	1	1		1			
1750-1999	6									1	2	1	1	1						
2000-2249	6									1		1	3		3					
2250-2499	7									3	1	3								
2500-2749	3										1			1			1			
2750-2999	2														1			1		
3000-3499	6															2	2	2		
3500-3999	4												1				2		1	
4000-4999	3											1			1			1		
5000 & Over	7															1	1	1	2	2
Totals	599	3	21	77	115	103	78	72	32	28	15	15	12	3	5	3	7	5	3	2

Table A-4
Pair Occurrences as a Percentage of Total Individual Descriptor Usage,
599 Most Common DDC Descriptors

f = Number of Pair Occurrences. F = Total Usages of Descriptor

f/F	No. Pairs	%	Cum. Pairs	Cum. %	f/F	No. Pairs	%	Cum. Pairs	Cum. %
.01	63,518	64.41	63,518	64.41	.36	22	.02	98,385	99.77
.02	12,508	12.68	76,026	77.10	.37	17	.02	98,402	99.79
.03	6,871	6.97	82,897	84.06	.38	17	.02	98,419	99.80
.04	4,195	4.25	87,092	88.32	.39	10	.01	98,429	99.81
.05	2,630	2.67	89,722	90.93	.40	18	.02	98,447	99.83
.06	1,802	1.83	91,524	92.81	.41	13	.01	98,460	99.85
.07	1,247	1.26	92,771	94.08	.42	13	.01	98,473	99.86
.08	1,012	1.03	93,783	95.10	.43	13	.01	98,486	99.87
.09	724	.73	94,507	95.84	.44	8	.01	98,494	99.88
.10	615	.62	95,122	96.46	.45	12	.01	98,506	99.89
.11	499	.51	95,621	96.97	.46	8	.01	98,514	99.90
.12	411	.42	96,032	97.38	.47	11	.01	98,525	99.91
.13	346	.35	96,378	97.73	.48	14	.01	98,539	99.93
.14	262	.27	96,640	98.00	.49	12	.01	98,551	99.94
.15	238	.24	96,878	98.24	.50	7	.01	98,558	99.95
.16	223	.23	97,001	98.47	.51	8	.01	98,566	99.95
.17	147	.15	97,148	98.62	.52	6	.01	98,572	99.96
.18	141	.14	97,389	98.76	.53	3	*	98,575	99.96
.19	147	.15	97,536	98.91	.54	5	.01	98,580	99.97
.20	126	.13	97,662	99.04	.55	2	*	98,582	99.97
.21	93	.09	97,755	99.13	.56	5	.01	98,587	99.97
.22	90	.09	97,845	99.22	.57	3	*	98,590	99.98
.23	73	.07	97,918	99.30	.58	–	–	–	99.98
.24	68	.07	97,986	99.37	.59	4	*	98,594	99.98
.25	54	.05	98,040	99.42	.60	2	*	98,596	99.98
.26	55	.06	98,095	99.48	.61	2	*	98,598	99.99
.27	40	.04	98,135	99.52	.62	2	*	98,600	99.99
.28	38	.04	98,173	99.55	.63	4	*	98,604	99.99
.29	31	.03	98,204	99.59	.64	3	*	98,607	99.99
.30	35	.04	98,239	99.62					
.31	30	.03	98,269	99.65	.71	1	*	98,608	99.99
.32	29	.03	98,298	99.68	.72	1	*	98,609	99.99
.33	31	.03	98,329	99.71	.73	1	*	98,610	99.99
.34	20	.02	98,349	99.73	.75	1	*	98,611	99.99
.35	14	.01	98,363	99.75	.86	1	*	98,612	100.00

* - Less than 0.005%

Table A-5
Pair Occurrences of the 50 Most Frequently Used DDC Descriptors
(Selected Summary Data)

Descriptor (Frequency of Usage Sequence)	No. of Doc.	No. of Pairs		% of Descr. Used With	599 Most Common Descr.				Remaining 4941 Descriptors			
		Diff.	Total		No. Used With	Total Pairs	% Used With	Average Pair Occur.	No. Used With	Total Pairs	% Used With	Average Pair Occur.
Design	6193	2669	26364	46.2%	571	18117	21.4%	31.7	2098	8247	42.5	3.9
Test	5237	2642	22289	47.7	579	14970	21.9	25.9	2063	7319	41.8	3.5
Mathematical Analysis	2470	1472	11424	30.2	509	8280	30.4	16.3	1163	3144	23.5	2.7
Measurement	1778	1846	9205	33.3	542	6034	27.4	11.1	1304	3171	26.4	2.4
Guided Missiles	1701	1125	10457	20.3	444	8599	39.5	19.4	681	1858	13.8	2.7
Temperature	1489	1568	7885	28.3	491	5410	31.3	11.0	1077	2475	21.8	2.3
Airborne	1380	990	6978	17.9	385	5391	38.9	14.0	605	1587	12.2	2.6
Production	1212	1239	5100	22.4	418	3368	33.7	8.1	821	1732	16.6	2.1
Theory	1209	1427	5839	25.8	492	4033	34.5	8.2	935	1806	18.9	1.9
Materiels	1155	1311	5790	23.7	443	3891	34.1	8.8	868	1899	17.6	2.2
Analysis	1113	1410	5035	25.5	505	3505	35.8	6.9	905	1530	18.3	1.7
Surface-to-Surface	1084	715	5557	12.9	362	4766	50.6	13.2	353	791	7.1	2.2
Great Britain	1075	1240	4704	22.4	466	3358	37.6	7.2	774	1346	15.7	1.7
Stability	1041	1176	5484	21.2	437	3390	37.2	9.1	739	1494	15.0	2.0
Effectiveness	1040	1305	4902	23.6	450	3335	34.5	7.4	855	1567	17.3	1.8
Flight Testing	929	708	4728	12.8	313	3746	44.2	12.0	395	982	8.0	2.5
Radar Equipment	915	658	5582	11.9	279	4409	42.4	15.8	379	1173	7.7	3.1
Instrumentation	908	1174	4741	21.2	445	3328	37.9	7.5	729	1413	14.8	1.9
Test Methods	868	1219	4045	22.0	464	2799	38.1	6.0	755	1246	15.3	1.7
Countermeasures	846	985	4528	17.8	398	3264	40.4	8.2	587	1264	11.9	2.2
Pressure	827	1016	4618	18.3	383	3309	37.7	8.6	633	1309	12.8	2.1
Detection	785	982	4188	17.7	368	2870	37.5	7.8	614	1318	12.4	2.1
Mechanical Properties	692	690	3485	12.5	271	2399	39.3	8.9	419	1086	8.5	2.6
Test Equipment	681	1022	3455	18.4	436	2554	42.7	5.9	586	901	11.9	1.5
Control Systems	673	753	3507	13.6	338	2637	44.9	7.8	415	870	8.4	2.1
Processing	645	774	2894	14.0	326	1992	42.1	6.1	448	902	9.1	2.0
Synthesis	622	774	3326	14.0	274	1994	35.4	7.3	500	1332	10.1	2.7
Physical Properties	601	1002	3239	18.1	362	2076	36.1	5.7	640	1163	13.0	1.8
Physiology	594	755	2809	13.6	196	1193	26.0	6.1	559	1616	11.3	2.9
Heat Transfer	591	731	3019	13.2	320	2239	43.8	7.0	411	780	8.3	1.9
Circuits	589	698	2996	12.6	275	2147	39.4	7.8	423	849	8.6	2.0
Determination	579	1071	2845	19.3	410	1744	38.3	4.3	661	1101	13.4	1.7
Chemical Reactions	573	743	2844	13.4	251	1669	33.8	6.6	492	1175	10.0	2.4
Surface-to-Air	569	406	2770	7.3	232	2372	57.1	10.2	174	398	3.5	2.3
Stresses	560	619	2707	11.2	282	1989	45.6	7.1	337	718	6.8	2.1
Polymers	560	732	3327	13.2	283	2266	38.7	8.0	449	1061	9.1	2.4
Projectiles	533	506	2937	9.1	245	2069	48.4	8.4	261	868	5.3	3.3
Aerodynamics	532	614	3077	11.1	285	2336	46.4	8.2	329	741	6.7	2.3
Combustion	523	584	2844	10.5	250	1998	42.8	8.0	334	846	6.8	2.5
Jet Planes	519	484	3072	8.7	238	2417	49.2	10.2	246	655	5.0	2.7
Radiation Effects	515	840	2744	15.2	318	1735	37.9	5.5	522	1009	10.6	1.9
Rocket Motors	492	590	3082	10.6	298	2398	50.5	8.0	292	684	5.9	2.3
Guidance	490	492	3141	8.9	269	2585	54.7	9.6	223	556	4.5	2.5
Reliability	489	636	2721	11.5	307	2177	48.3	7.1	329	544	6.7	1.7
Propagation	479	502	3000	9.1	227	2321	45.2	10.2	275	679	5.6	2.5
Solid Rocket Propellants	479	599	2617	10.8	302	2004	50.4	6.6	297	613	6.0	2.1
Scattering	452	638	2343	11.5	293	1630	45.9	5.6	345	713	7.0	2.1
Model Tests	451	505	2399	9.1	237	1759	46.9	7.4	268	640	5.4	2.4
Statistical Analysis	444	721	1882	13.0	309	1197	42.9	3.9	412	685	8.3	1.7
Vibration	443	698	2155	12.6	331	1524	47.4	4.6	367	631	7.4	1.7
Totals	-	48256	250680	-	17909	178193	-	9.9	30347	72487	-	2.4

Source: Sample of 38,402 DDC Documents

Table A-6
Summary Statistics of 599 Most Common DDC Descriptors,
Classified by Field of Interest

Major Field	No. Groups		Descriptors				Pair Permutations				Ave. Occur. Pair
	Total	In 599	Total	In 599	% Total	% 599	No. Diff.	%	Total Occur.	%	
Aeronautics	19	11	338	70	4.8	11.7	11,415	11.6	65,843	13.2	5.77
Applied Research & Military Aspects	21	14	443	37	6.3	6.2	6,137	6.2	37,919	7.6	6.18
Bio-Sciences	19	4	350	6	5.0	1.0	521	0.5	2,814	0.6	5.40
Chemistry	32	12	859	57	12.2	9.5	8,476	8.6	33,411	6.7	3.94
Civil Engineering	4	1	57	3	0.8	0.5	473	0.5	1,748	0.3	3.70
Electronic & Electrical Engineering	31	16	601	56	8.6	9.4	9,118	9.2	50,971	10.3	5.59
Geophysical Sciences & Geography	11	5	446	21	6.4	3.5	3,661	3.7	13,716	2.7	3.75
Human Engineering & Psychology	9	5	165	17	2.4	2.8	1,737	2.0	6,267	1.3	3.61
Industrial Methods & Processes	6	2	166	14	2.4	2.3	2,337	2.4	11,279	2.3	4.83
Materials and Metals	19	15	425	44	6.1	7.4	6,468	6.5	28,453	5.7	4.40
Mechanical & Automotive Engineering	14	6	283	9	4.0	1.5	1,617	1.6	7,843	1.6	4.85
Medicine	24	8	789	17	11.3	2.8	1,794	1.8	7,430	1.5	4.14
Miscellaneous Arts & Sciences	11	7	278	43	4.0	7.2	10,637	10.8	80,976	16.3	7.61
Ordnance	17	13	424	45	6.0	7.5	6,251	6.3	27,765	5.6	4.44
Physics and Mathematics	27	23	896	133	12.8	22.2	23,126	23.4	98,084	19.7	4.24
Propulsion Systems & Power Plants	13	6	180	10	2.6	1.7	1,808	1.8	9,871	2.0	5.46
Research Facilities & Instrumentation	8	4	166	6	2.4	1.0	1,272	1.3	5,823	1.2	4.58
Ships & Marine Equipment	3	1	106	2	1.5	0.3	301	0.3	822	0.2	2.73
Space Technology	4	2	32	9	0.4	1.5	1,463	1.5	5,815	1.2	3.97
Totals	292	155	7004	599	100.0	100.0	98,612	100.0	496,850	100.0	5.04

Table A-7
Summary of Pair Associations Classified by DDC Groups to Which Descriptors are Assigned
(Sample Page)

FIELD 10
GROUP 145

FD GRP	NUM	DIFF	OCCUR	FD GRP	NUM	DIFF	OCCUR	FD GRP	NUM	DIFF	OCCUR	FD GRP	NUM	DIFF	OCCUR	FD GRP	NUM	DIFF	OCCUR
01 005	004	0014	00025	01 006	005	0045	00133	01 008	001	0001	00008	01 009	003	0005	00064	01 010	002	0002	00008
01 012	004	0004	00043	01 041	001	0001	00002	01 094	001	0001	00003	01 114	004	0010	00081	01 255	001	0001	00001
02 002	002	0002	00002	07 062	002	0006	00069	02 067	001	0002	00004	02 078	004	0005	00030	02 102	002	0004	00011
02 127	002	0002	00014	02 170	001	0001	00004	02 180	001	0001	00004	02 183	002	0002	00013	02 184	001	0001	00004
02 196	001	0001	00011	02 226	001	0001	00002	02 227	003	0012	00021	03 036	001	0001	00002	03 086	001	0001	00001
03 098	001	0002	00011	04 063	003	0003	00014	04 018	001	0001	00003	04 049	005	0035	00132	04 051	002	0005	00018
04 053	005	0018	00069	04 087	001	0001	00008	04 106	003	0008	00059	04 131	005	0024	00199	04 157	005	0011	00061
04 190	002	0001	00015	04 193	002	0002	00005	04 207	002	0003	00005	05 061	003	0004	00019	06 022	003	0005	00019
06 025	003	0004	00011	06 027	005	0026	00087	06 059	003	0006	00011	06 074	004	0005	00010	06 075	004	0007	00026
06 079	005	0010	00052	06 081	003	0007	00026	06 082	004	0012	00076	06 166	002	0002	00002	06 197	001	0001	00002
06 215	001	0003	00015	06 232	001	0001	00003	06 273	002	0002	00018	06 274	002	0002	00004	07 054	002	0009	00023
07 108	004	0009	00034	07 181	001	0001	00007	08 001	003	0003	00024	08 119	001	0001	00001	08 202	001	0002	00007
08 223	001	0001	00001	08 270	001	0002	00002	09 217	005	0041	00421	09 286	002	0002	00013	10 004	003	0006	00039
10 016	005	0020	00140	10 056	004	0006	00068	10 093	003	0003	00036	10 099	004	0022	00086	10 112	005	0005	00040
10 145	005	0016	00212	10 146	004	0007	00031	10 158	004	0009	00049	10 159	002	0003	00013	10 160	005	0005	00051
10 164	002	0002	00002	10 182	002	0003	00012	10 212	004	0009	00110	10 239	003	0007	00041	11 092	001	0001	00007
11 216	005	0005	00016	11 240	002	0004	00023	11 256	001	0001	00016	11 275	003	0004	00015	12 023	002	0002	00002
12 150	002	0002	00004	12 209	005	0009	00041	12 269	001	0001	00003	13 060	002	0008	00013	13 071	005	0010	00038
13 073	002	0003	00011	13 104	003	0006	00020	13 246	001	0001	00003	13 292	005	0045	00784	14 020	005	0015	00072
14 026	002	0002	00006	14 032	004	0007	00021	14 042	001	0001	00001	14 048	001	0002	00016	14 090	004	0010	00051
14 100	002	0007	00007	14 115	003	0007	00015	14 165	001	0001	00001	14 237	002	0003	00005	14 261	001	0001	00002
14 296	002	0003	00006	15 029	004	0012	00030	15 065	003	0003	00003	15 066	005	0021	00108	15 076	005	0037	00254
15 105	005	0012	00093	15 116	005	0007	00052	15 117	005	0031	00275	15 121	003	0005	00009	15 122	004	0004	00018
15 133	002	0002	00003	15 147	005	0013	00046	15 148	004	0013	00105	15 178	003	0004	00018	15 186	001	0001	00003
15 187	005	0028	00100	15 200	002	0003	00009	15 211	004	0008	00013	15 225	002	0003	00004	15 230	004	0009	00024
15 247	005	0036	00214	15 249	004	0007	00025	15 287	003	0004	00006	16 057	004	0010	00029	16 085	004	0005	00030
16 089	002	0002	00009	16 123	001	0001	00009	16 173	004	0004	00055	16 219	004	0005	00018	17 080	001	0001	00001
17 136	002	0002	00006	17 208	001	0001	00013	17 234	005	0008	00037	18 244	003	0004	00018	19 251	003	0008	00017
19 253	004	0009	00021																

FIELD 01	0084	00368	FIELD 02	0040	00189	FIELD 03	0004	00014	FIELD 04	0115	00582	FIELD 05	0004	00019
FIELD 06	0093	00362	FIELD 07	0019	00064	FIELD 08	0009	00035	FIELD 09	0043	00434	FIELD 10	0123	00932
FIELD 11	0015	00077	FIELD 12	0014	00050	FIELD 13	0073	00869	FIELD 14	0059	00203	FIELD 15	0263	01412
FIELD 16	0027	00150	FIELD 17	0012	00057	FIELD 18	0004	00018	FIELD 19	0017	00038			

TOTALS OF 145 GROUP 1018 05873

Table A-8

599 Most Common DDC Descriptors: Occurrences of Pair Associations Within One Group or One Field-of-Interest

DDC Field	599 Descriptors				Pairs Within Groups				Pairs Within Fields			
	Desc.	Grps.	Descr. Pairs	Pair Occur.	No. Pairs	No. Occur.	% Pairs	% Occur.	No. Pairs	No. Occur.	% Pairs	% Occur.
Aeronautics	70	11	11,415	65,843	520	4,262	4.8	6.9	1,446	11,929	14.5	22.1
Applied Research & Mil. Aspects	37	14	6,137	37,919	60	1,413	1.0	3.9	374	4,264	6.5	12.7
Bio-Science	6	4	521	2,814	3	63	0.6	2.3	15	192	3.0	7.3
Chemistry	57	12	8,476	33,411	118	873	1.4	2.7	959	5,714	12.8	20.6
Civil Engineering	3	1	473	1,748	3	17	0.6	1.0	3	17	0.6	1.0
Electronic/Electrical Eng.	56	16	9,118	50,971	149	1,578	1.7	3.2	970	6,356	11.9	14.2
Geophysical Sciences	21	5	3,661	13,716	60	573	1.7	4.4	140	809	4.0	6.3
Human Eng. & Psychology	17	5	1,737	6,267	33	377	1.9	6.4	108	720	6.6	13.0
Industrial Methods	14	2	2,337	11,279	60	678	2.6	6.4	71	766	3.1	7.3
Materials & Metals	44	15	6,468	28,453	66	849	1.0	3.1	466	2,858	7.8	11.2
Mech/Auto. Engineering	9	6	1,617	7,843	3	26	0.2	0.3	20	108	1.3	1.4
Medicine	17	8	1,794	7,430	18	159	1.0	2.2	98	1,058	5.6	16.6
Misc. Arts & Sciences	43	7	10,637	80,976	171	4,876	1.6	6.4	591	7,791	5.9	10.6
Ordnance	45	13	6,251	77,765	92	1,858	1.5	7.2	564	4,472	9.9	19.2
Physics & Mathematics	133	23	23,126	98,084	409	3,452	1.8	3.6	3,673	17,772	18.9	22.1
Propulsion Systems	10	6	1,808	9,871	5	85	0.3	0.9	38	624	2.1	6.7
Research Facilities	6	4	1,272	5,823	2	16	0.2	0.3	11	81	0.9	1.4
Ships & Marine Equipment	2	1	301	822	1	5	0.3	0.6	1	5	0.3	0.6
Space Technology	9	2	1,463	5,815	15	155	1.0	2.7	34	404	2.4	7.5
Totals	599	155	98,612	495,850	1,788	21,315	1.8	4.5	9,582	65,940	10.8	15.3

Table A-9

Pair Associations of 599 Most Common DDC Descriptors, Classified by Field-of-Interest Assignment of Each Member of Pair

A: Number of Different Pairs Occurring

Fld.	DDC Field of Interest	Descr. in Field	No. Diff. Pairs	Pairs Within Groups	Number of Different Pairs with Descriptors in Fields:* 01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01	Aeronautics	70	9,969	520	1146	650	9	425	54	812	417	164	126	488	217	74	1238	733	2406	276	142	29	261
02	Applied Research	37	5,763	60	650	374	29	216	50	989	264	132	91	210	87	90	818	453	977	81	95	27	130
03	Bio-Sciences	6	506	3	9	29	15	90		8	18	21	24	25	6	67	58	16	113		4	2	1
04	Chemistry	57	7,517	118	425	214	90	259	7	315	214	65	340	988	111	209	647	321	2262	225	75	13	35
05	Civil Engineering	3	470	3	54	50		7	3	64	13	15	9	33	17	3	75	46	52	6	9	5	4
06	Electronic/Electrical Eng.	56	8,148	149	812	989	8	315	64	970	340	110	142	262	110	77	1175	457	1899	98	147	34	239
07	Geophysical Sciences	21	3,521	60	417	264	18	214	18	340	140	60	58	156	51	71	402	201	941	38	99	12	62
08	Human Eng. and Psychology	17	1,629	33	164	132	21	65	15	110	60	108	20	33	36	100	321	95	267	16	26	10	30
09	Industrial Methods	14	2,266	60	126	91	24	340	9	142	58	20	71	346	39	48	215	144	497	53	22	10	11
10	Materials and Metals	44	6,002	66	488	210	25	338	33	262	158	33	346	460	140	66	577	395	1474	180	77	19	55
11	Mech./Auto. Engineering	9	1,597	3	217	87	6	111	17	110	51	36	39	140	20	29	170	160	328	30	20	9	17
12	Medicine	17	1,696	28	74	90	67	209	3	77	71	100	48	66	29	53	201	82	434	14	14	4	15
13	Misc. Arts and Sciences	43	10,046	171	1238	813	58	647	73	1175	402	321	215	577	170	201	591	833	2154	179	147	36	174
14	Ordnance	45	5,687	92	733	452	16	321	46	457	201	95	144	395	160	82	833	364	916	109	75	17	70
15	Physics and Mathematics	132	19,453	409	2406	977	113	2262	52	1899	941	267	497	1474	328	434	2189	916	3673	363	278	65	319
16	Propulsion Systems	10	1,770	5	276	81		225	6	98	38	16	53	180	30	14	179	109	363	36	31	3	30
17	Research Facilities	6	1,261	2	142	95	4	75	9	147	39	26	22	77	20	14	147	75	278	31	11	3	26
18	Ships and Marine Eqpt.	2	300	1	29	27	2	13	5	34	12	10	10	19	9	4	36	17	65	3	3	1	1
19	Space Technology	9	1,429	15	266	130	1	35	4	139	62	30	11	65	17	15	174	70	319	30	26	1	34

B: Total Occurrences of Pairs

Fld.	DDC Field of Interest	Descr. in Field	Total Pair Occur.	Occur. Within Groups	Number of Occurrences with Descriptors in Fields:* 01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01	Aeronautics	70	53,914	4262	11929	4452	33	1084	213	4978	1620	387	488	1627	1805	146	9247	2627	9600	1521	680	91	1386
02	Applied Research	37	33,655	1413	4452	4264	253	678	151	6990	896	401	345	680	350	329	7220	1853	3093	286	658	105	351
03	Bio-Sciences	6	2,622	63	33	253	192	311		9	56	190	122	77	10	657	139	35	524		8	5	1
04	Chemistry	57	27,697	873	1084	678	311	5714	9	990	475	161	1259	4237	262	483	2586	832	7501	919	129	23	44
05	Civil Engineering	3	1,731	17	213	151		9	17	265	39	59	20	80	73	3	546	107	83	16	31	10	4
06	Electronic/Electrical Eng.	56	44,615	1578	4978	6990	9	990	265	6356	1266	252	535	823	341	170	11505	1363	7005	543	694	78	452
07	Geophysical Sciences	21	12,907	573	1620	896	56	475	39	1266	809	123	169	333	194	185	2425	585	3187	102	275	16	152
08	Human Eng. and Psychology	17	5,547	377	387	401	190	161	59	252	123	730	97	31	131	316	1466	171	793	51	42	25	50
09	Industrial Methods	14	10,513	678	488	345	122	1259	20	535	169	97	766	1953	239	121	1586	467	2083	170	67	14	12
10	Materials and Metals	44	25,595	849	1627	680	77	4237	80	823	333	31	1953	2858	523	123	3439	949	6040	1394	184	49	125
11	Mech./Auto. Engineering	9	7,735	26	1805	350	10	262	73	341	194	131	239	523	106	54	1267	504	1581	180	51	12	45
12	Medicine	17	6,372	159	146	329	657	483	3	170	185	316	121	123	54	1958	790	335	1489	26	29	4	34
13	Misc. Arts and Sciences	43	73,185	4876	9247	7220	139	2586	546	11505	2425	1466	1586	3439	1267	790	7791	5663	13597	3394	1478	174	362
14	Ordnance	45	23,293	1858	2627	1853	35	832	107	1363	585	171	467	949	504	335	5663	4472	2590	372	231	37	98
15	Physics and Mathematics	133	80,312	3452	9600	3093	524	7501	83	7005	3187	793	2083	6040	1581	1489	13587	2590	27772	1397	827	156	1003
16	Propulsion Systems	10	9,247	85	1521	286		919	16	543	102	51	170	1394	180	26	3394	372	1397	634	119	6	125
17	Research Facilities	6	5,742	16	680	658	8	129	31	694	275	42	67	184	51	29	1478	231	827	119	91	6	152
18	Ships and Marine Eqpt.	2	817	5	91	105	5	23	10	78	16	25	14	49	12	4	174	37	156	6	6	5	1
19	Space Technology	9	5,411	155	1386	351	1	44	4	452	152	50	12	125	45	34	362	98	1003	125	152	1	404

*Duplications eliminated when both descriptors of a pair are in one field-of-interest

18	Ships and Marine Eqpt.	2	817	5	91	105	5	23	10	76	16	25	14	49	22	4	174	37	156	9	6	2	1
19	Space Technology	9	5,411	155	1386	351	1	44	4	452	152	60	13	125	45	34	762	98	1003	135	152	1	404

*Duplications eliminated when both descriptors of a pair are in one field-of-interest.

Table A-9

Pair Associations of 599 Most Common DDC Descriptors, Classified by Field-of-Interest Assignment of Each Member of Pair

C: Different Pairs as a Percentage of Maximum Possible Number

Fld.	DDC Field of Interest	Descr. in Field	Field Avg. %	Number of Different Pairs as Percentage of Maximum Possible in Fields:*																		
				01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
01	Aeronautics	70	25	60	29	2	11	26	21	28	14	13	16	34	6	43	23	26	39	34	21	42
02	Applied Research	37	27	25	56	13	10	45	48	34	21	18	13	26	14	51	27	20	22	43	56	39
03	Bio-Sciences	6	14	2	13	100	26	--	2	14	21	29	9	11	66	22	6	14	-	11	17	2
04	Chemistry	57	23	11	10	26	57	4	10	18	7	43	33	22	22	26	13	30	79	22	31	7
05	Civil Engineering	3	26	26	45	-	4	100	38	29	29	21	25	63	6	58	34	13	20	50	53	15
06	Electronic/Electrical Eng.	56	26	21	48	2	10	38	63	29	12	18	11	32	8	49	18	25	18	44	30	28
07	Geophysical Sciences	21	29	28	34	14	18	29	29	67	17	20	17	27	20	45	21	34	28	47	29	33
08	Human Eng. & Psychology	17	16	14	21	21	7	29	12	17	79	8	4	24	35	44	12	12	9	25	25	20
09	Industrial Methods	14	27	13	18	29	43	21	18	20	8	78	56	31	20	36	23	27	38	26	36	9
10	Materials and Metals	44	24	15	13	9	39	25	11	17	4	56	49	35	9	30	20	25	41	29	22	16
11	Mech./Auto. Engineering	9	30	24	25	11	22	63	22	27	24	31	35	56	19	44	40	27	33	37	59	21
12	Medicine	17	17	6	14	66	22	6	8	20	35	20	9	19	72	27	11	19	8	14	12	10
13	Misc. Arts and Sciences	43	40	41	51	22	26	58	49	45	44	36	30	44	27	63	43	38	42	57	42	45
14	Ordnance	45	22	23	27	6	13	34	18	21	12	23	20	40	11	43	52	15	24	28	19	17
15	Physics and Mathematics	133	27	26	20	14	30	13	25	34	12	27	25	27	19	38	15	42	27	35	24	27
16	Propulsion Systems	10	30	39	22	-	39	20	18	18	9	38	41	33	8	42	24	27	84	52	15	33
17	Research Facilities	6	35	34	43	11	22	50	44	47	25	26	29	37	14	57	28	35	52	73	25	48
18	Ships and Marine Eqpt.	2	25	21	36	17	11	83	30	29	29	36	22	50	12	42	19	24	15	25	100	6
19	Space Technology	9	27	42	39	2	7	15	28	33	20	9	16	21	10	45	17	22	33	48	6	94

*Duplications eliminated when both descriptors of a pair are in one field-of-interest.

Table A-10
Number of Descriptor Pair Associations AB,
Classified by Frequency of Usage of Descriptor A

Descr. Occ.	No. of Descr.	Diff. Pairs	Total Pairs	Average Pair Occ.	Ave. Prs. per Descr.
1	857	4,791	4,791	1.00	5.6
2	570	5,867	6,309	1.08	10.3
3	416	6,273	7,092	1.13	15.1
4	317	6,072	7,297	1.20	19.2
5	230	5,085	6,368	1.25	22.1
6	197	5,251	6,751	1.29	26.7
7	163	4,726	6,240	1.32	29.0
8	145	4,765	6,394	1.34	32.9
9	155	5,736	7,834	1.37	37.0
10	119	4,756	6,621	1.39	40.0
11	109	4,539	6,657	1.47	41.6
12	112	5,292	7,477	1.41	47.3
13	83	3,902	5,824	1.49	47.0
14	68	3,396	5,154	1.52	49.9
15	83	4,600	6,919	1.50	55.4
16	65	3,619	5,785	1.60	55.7
17	48	2,858	4,641	1.62	59.5
18	61	4,012	6,350	1.58	65.8
19	46	3,123	5,047	1.62	67.9
20-24	241	17,633	29,725	1.68	73.2
25-29	163	14,189	24,122	1.70	87.0
30-34	147	14,752	26,177	1.77	100.4
35-39	134	14,319	27,212	1.90	106.9
40-44	87	10,273	20,193	1.97	118.1
45-49	90	12,032	24,661	2.05	133.7
50-59	121	17,345	36,282	2.09	143.3
60-69	99	16,060	35,984	2.24	162.2

Descr. Occ.	No. of Descr.	Diff. Pairs	Total Pairs	Average Pair Occ.	Ave. Prs. per Descr.
70-79	70	12,397	28,406	2.29	177.1
80-89	74	15,174	34,995	2.31	205.1
90-99	44	9,656	23,417	2.43	219.5
100-124	100	25,061	63,012	2.52	250.6
125-149	60	16,402	45,244	2.76	273.4
150-174	47	14,750	41,517	2.81	313.8
175-199	38	13,180	37,918	2.88	346.8
200-224	35	13,019	40,045	3.08	372.0
225-249	26	11,084	33,052	2.98	426.3
250-274	18	7,166	25,344	3.54	398.1
275-299	12	5,232	19,463	3.72	436.0
300-324	14	6,919	23,459	3.39	494.2
325-349	9	5,043	15,972	3.17	560.3
350-374	5	2,867	10,410	3.63	573.4
375-399	4	2,246	7,757	3.46	561.5
400-424	6	3,562	12,480	3.50	593.7
425-449	4	2,551	8,537	3.35	637.8
450-474	2	1,143	4,742	4.15	571.5
475-499	5	2,819	14,561	5.17	563.8
500-599	13	8,783	37,991	4.33	675.6
600-699	6	5,015	19,906	3.97	835.8
700-799	1	982	4,188	4.26	982.0
800-899	3	3,220	13,191	4.10	1,073.3
900-999	3	2,540	15,051	5.93	846.7
1000 & Up	15	22,335	137,013	6.13	1,489.0
Total	3540	418,412	1,061,588	2.54	75.5

Source: Sample of 38,402 DDC Documents

REFERENCES

[1] Univac Division of Sperry Rand Corp., "Optimization and Standardization of Information Retrieval Language and Systems," July 1962. AFOSR-3216, Final Report under Contr. AF49(638)-835.

[2] Prywes, N. S., et al, "The Multi-List System," November 1961. Moore School of Electrical Engineering, University of Pennsylvania. Technical Status Report No. 1 under Contr. NOnr 551(40).

[3] Lefkovitz, David, "Automatic Stratification of Descriptors," 15 September 1963. Moore School of Electrical Engineering, University of Pennsylvania. Technical Report under Contr. NOnr 551(40). (Moore School Report No. 64-03).

[4] Univac Division of Sperry Rand Corp., "Multi-List System: Preliminary Report of a Study Into Automatic Attribute Group Assignment," 27 March 1963. Tech. Status Report No. 1 under Contr. AF49(638)1194.

[5] Ibid., "Multi-List System: Additional Notes on a Study Into Automatic Attribute Group Assignment," 15 October 1963. Tech. Status Report No. 2 under Contr. AF49(638)1194.

[6] Ibid., "Multi-List System: Additional Notes on a Study Into Automatic Attribute Group Assignment," 27 March 1964. Tech. Status Report No. 3 under Contr. AF49(638)1194.

[7] Ibid., "Multi-List System: Final Notes on a Study Into Automatic Attribute Group Assignment," 1 October 1964. Tech. Status Report No. 4 under Contr. AF49(638)1194.

[8] Ibid., "Some Notes on the Use and Data Processing Aspects of Association Factors in IS&R Systems," 30 March 1965. Tech. Status Report No. 5 under Contr. AF49(638)1194.

[9] Black, D. V., and Patrick, R. L., "Index Files: Their Loading and Organization for Use," 1 May 1963. Planning Research Corp., Los Angeles, Calif.

[10] Maron, M. E., and Kuhns, J. L., "On Relevance, Probabilistic Indexing and Information Retrieval," Journal of the ACM, July 1960 (Vol. 7, No. 3), pp. 216-244.

[11] Stiles, H. Edmund, "The Association Factor in Information Retrieval." Journal of the ACM, April 1961 (Vol. 8, No. 2), pp. 271-279.

[12] Doyle, L. B., "Semantic Road Maps for Literature Searches," Journal of the ACM, October 1961 (Vol. 8, No. 5), pp. 553-578.

U. S. GOVERNMENT PRINTING OFFICE: 1965—766-929

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA — R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Sperry Rand Corp., Univac Division
Blue Bell, Pa. 19422

2A. REPORT SECURITY CLASSIFICATION
[X] Unclassified
Other — Specify

2B. GROUP

3. REPORT TITLE
Optimization and Standardization of Information Retrieval Language and Systems

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
☐ Scientific Report [X] Final Report ☐ Journal Article ☐ Proceedings ☐ Book

5. AUTHOR(S) (Last name, first name, initial)
Fossum, Earl G.
Kaskey, Gilbert

6. REPORT DATE AS PRINTED	7A. TOTAL NO. OF PAGES	7B. NO. OF REFS
28 Jan. 1966	87	12

8A. CONTRACT OR GRANT NO. AF 49(638)-1194
B. PROJECT NO. 9769
C. 61445014
D.

9A. ORIGINATOR'S REPORT NUMBER(S) (if given)

9B. OTHER REPORT NO.(S) (Any other numbers that may be assigned this report)
AFOSR 66-0628
AD

10. AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited.

[X] Available from DDC
[X] Available from CFSTI
☐ Available from Source
☐ Available Commercially

11. SUPPLEMENTARY NOTES (Citation)

12. SPONSORING MILITARY ACTIVITY
AF Office of Scientific Research (SRI)
Office of Aerospace Research
Washington, D. C. 20333

13 ABSTRACT
This report analyzes and evaluates methods of organizing data files, primarily for document retrieval applications. Three principal techniques are examined: the Multi-List System, the list-organized file, and the inverted and document-sequenced file. The terms assigned to 38,402 DDC documents constituted the data base. Statistical analyses were made of term associations based on 599 most common DDC descriptors. The Multi-List System is a variation of the conventional list-organized file in which chains are based on groups of two or three index terms rather than a single one. Results indicate the need of a large amount of processing against an extensive data base; since most documents have almost as many groups as index terms, the postulated reduction in lists traversing a given document cannot be realized. Analysis shows that the list-organized file is an amalgamation of the inverted and document-sequenced files, and that maintenance and use of the two separate files is more efficient when requirements cannot be met by the inverted file alone. A technique for optimizing organization of the two files to minimize actual computing and over-all elapsed processing times is described. The 599 descriptors had 49,306 different pair combinations, 41% of the pairs occurring only once and almost 80% of the pairs occurring five times or less. It is viewed as dubious that any particular significance can be attached to a unique index term "association." There appears potential value in using relationships implicit in the hierarchic structure of a thesaurus, both for processing search requests and to aid in assigning descriptors by such techniques as "lowest level indexing."

DD FORM 1473
1 JAN 64

UNCLASSIFIED
Security Classification